UNIVERZITET U BANJOJ LUCI
ELEKTROTEHNIČKI FAKULTET

# PRIMJENA TEHNIKA MAŠINSKOG UČENJA ZA PREDIKCIJU ZDRAVLJA LITIJUM-JONSKIH BATERIJA

**MASTER RAD**

Mentor:
prof. dr Vladimir Risojević

Kandidat:
Kristijan Stepanov

**Banja Luka, oktobar 2024.**


UNIVERSITY OF BANJA LUKA
FACULTY OF ELECTRICAL ENGINEERING


# APPLICATION OF MACHINE LEARNING TECHNIQUES FOR PREDICTING THE HEALTH OF LITHIUM-ION BATTERIES

**MASTER'S THESIS**

Mentor:
Dr. Vladimir Risojevic, Assoc. Prof.

Candidate:
Kristijan Stepanov

**Banja Luka, October 2024.**

# Informacije o komisiji i master radu

**Komisija:**
prof. dr Mitar Simić, predsjednik
prof. dr Vladimir Risojević, mentor
doc. dr Slavica Gajić, član

Univerzitet u Banjoj Luci, Elektrotehnički fakultet

**Naslov master rada:**
Primjena tehnika mašinskog učenja za predikciju zdravlja litijum-jonskih baterija.

**Rezime:**
Ovaj rad istražuje primjenu različitih algoritama mašinskog učenja za predikciju stanja zdravlja (SOH) litijum-jonskih baterija koristeći podatke iz NASA i *Toyota* skupova podataka. Fokus je stavljen na optimizaciju hiperparametara korištenjem grid pretrage i unakrsne validacije, kako bi se postigli što tačniji modeli. U radu su korišteni regresivni i klasifikacioni modeli, uključujući konvolucione neuronske mreže (CNN), potpuno povezane neuronske mreže (MLP), kao i napredne ansambl metode poput *CatBoost*-a i *XGBoost*-a. Poseban akcenat je stavljen na izazove u pripremi podataka, kao što su parsiranje i standardizovano pretprocesiranje, koje je primijenjeno nad podacima u skladu sa zahtjevima svakog algoritma. Rezultati pokazuju da su *CatBoost* modeli postigli najbolje performanse, s niskim vrijednostima srednje kvadratne greške (MSE) i visokim koeficijentom determinacije ($R^2$), što ukazuje na njihovu efikasnost u predikciji SOH-a. Ova istraživanja pružaju značajne uvide u optimizaciju modela za predikciju performansi baterija, što može biti od koristi u daljim istraživanjima i razvoju u oblasti upravljanja baterijama.

**Ključne riječi:** stanje zdravlja baterije, litijum-jonske baterije, mašinsko učenje, optimizacija hiperparametara, grid pretraga, srednja kvadratna greška.

**Naučna oblast:** inženjerstvo i tehnologija

**Naučno polje**: elektrotehnika, elektronika i informacione tehnologije

**Klasifikaciona oznaka:** P176, T140

Information about the committee and the master's thesis

**Committee:**
Dr. Mitar Simić, Assoc. Prof., Chair
Dr. Vladimir Risojevic, Assoc. Prof., Mentor
Dr. Slavica Gajić, Asst. Prof., Member

University of Banja Luka, Faculty of Electrical Engineering

**Title of the Master's Thesis:**
Application of Machine Learning Techniques for Predicting the Health of Lithium-Ion Batteries.

**Abstract:**
This thesis explores the application of various machine learning algorithms for predicting the state of health (SOH) of lithium-ion batteries using data from NASA and Toyota datasets. The focus is on hyperparameter optimization using grid search and cross-validation to achieve the most accurate models. The study employs both regression and classification models, including convolutional neural networks (CNN), multilayer perceptron (MLP), and advanced ensemble methods such as CatBoost and XGBoost. Special emphasis is placed on the challenges in data preparation, such as parsing and standardized preprocessing, which was applied to the data according to the requirements of each algorithm. The results indicate that CatBoost models achieved the best performance, with low mean squared error (MSE) values and high coefficients of determination ($R^2$), demonstrating their effectiveness in SOH prediction. This research provides significant insights into model optimization for battery performance prediction, which can be beneficial for further research and development in battery management systems.

**Keywords:** state of health, lithium-ion batteries, machine learning, hyperparameter optimization, grid search, mean squared error.

**Scientific Field:** Engineering and Technology

**Discipline**: Electrical Engineering, Electronics, and Information Technology

**Classification Code:** P176, T140

# SADRŽAJ

**PRIMJENA TEHNIKA MAŠINSKOG UČENJA ZA PREDIKCIJU ZDRAVLJA LITIJUM-JONSKIH BATERIJA ..... 1**

**1. UVOD ..... 7**

1.1. MOTIVACIJA ..... 7
1.2. POSTOJEĆI RADOVI ..... 8
1.3. PROBLEMI ..... 11
1.4. DOPRINOSI ..... 11
1.5. PREGLED SADRŽAJA ..... 12

**2. POJAM ZDRAVLJA LITIJUM-JONSKE BATERIJE ..... 13**

2.1. UVOD I ISTORIJSKI PREGLED ..... 13
2.2. SOH METRIKA ..... 13
2.3. METRIKA PREOSTALOG VIJEKA TRAJANJA BATERIJE ..... 14

**3. OPIS KORIŠTENIH ALGORITAMA ..... 16**

3.1. REGRESIVNE METRIKE ..... 16
*3.1.1. Srednja kvadratna greška ..... 16*
*3.1.2. Koeficijent determinacije (R-kvadrat) ..... 17*
3.2. KLASIFIKACIONE METRIKE ..... 17
*3.2.1. Matrica konfuzija ..... 17*
*3.2.2. Tačnost ..... 18*
*3.2.3. Preciznost ..... 18*
*3.2.4. Odziv ..... 18*
*3.2.5. F1 metrika ..... 19*
*3.2.6. Primjer matrice konfuzije i analiza ..... 19*
3.3. KLASIČNI ALGORITMI MAŠINSKOG UČENJA ..... 19
*3.3.1. Metoda potpornih vektora ..... 21*
*3.3.2. Linearna regresija ..... 22*
*3.3.3. Ridge ..... 23*
*3.3.4. Lasso ..... 23*
*3.3.5. Slučajna šuma ..... 24*
*3.3.6. XGBoost ..... 24*
*3.3.7. CatBoost ..... 26*
3.4. ALGORITMI DUBOKOG UČENJA ..... 27
*3.4.1. Potpuno povezana neuronska mreža ..... 29*
*3.4.2. Rekurentna neuronska mreža ..... 30*
*3.4.3. Konvoluciona neuronska mreža ..... 32*

**4. KORIŠTENI SKUPOVI PODATAKA ..... 34**

4.1. NASA SKUP PODATAKA ..... 34
*4.1.1. Podjela i pretprocesiranje podataka ..... 36*
4.2. *TOYOTA* SKUP PODATAKA ..... 36
*4.2.1. Pretprocesiranje podataka ..... 39*

**5. EKSPERIMENTALNI REZULTATI ..... 41**

5.1. PRIMJENA REGRESIJE SA NASA SKUPOM PODATAKA ..... 43
5.2. EFEKAT VREMENSKE INFORMACIJE NA PREDIKCIJU UNUTAR CIKLUSA ..... 47
*5.2.1. Izazovi ..... 47*
*5.2.2. Rezultati eksperimenta ..... 48*
5.3. PRIMJENA REGRESIJE SA *TOYOTA* SKUPOM PODATAKA ..... 48
5.4. PRIMJENA KLASIFIKACIJE SA *TOYOTA* SKUPOM PODATAKA ..... 51
5.5. PRIMJENA PRENOSNOG UČENJA ZA KLASIFIKACIJU KORIŠTENJEM REGRESIONOG MODELA ..... 58
5.6. EFEKAT AUGMENTACIJE PODATAKA ..... 61
*5.6.1. Gausov šum ..... 64*
*5.6.2. Interpolacija ..... 66*
5.7. KOMPARATIVNA ANALIZA ..... 68

**6. ZAKLJUČAK I BUDUĆI RAD ..... 71**

**LITERATURA ..... 72**

# 1. Uvod

Tehnologije bazirane na litijum-jonskim baterijama su postale neizostavan dio modernog društva, omogućavajući napajanje širokog spektra uređaja i sistema, od prenosivih elektronskih uređaja do velikih energetskih sistema u industriji obnovljivih izvora energije. Razvoj i primjena ovih baterija direktno su povezani sa njihovom sposobnošću da pružaju pouzdano i dugotrajno snabdjevanje energijom. Međutim, kao i svi tehnički sistemi, litijum-jonske baterije su podložne procesu degradacije, što može značajno uticati na njihovu dugovječnost i pouzdanost.

Predviđanje stanja zdravlja (eng. *state of health*; SOH) baterije predstavlja ključan aspekt u upravljanju njihovim performansama tokom vremena. Tačna predviđanja mogu omogućiti optimizovano korištenje resursa, smanjenje troškova, i povećanje ukupne efikasnosti sistema u kojima se ovi energetski izvori koriste. S obzirom na širenje upotrebe litijum-jonskih baterija u sve više sektora, potreba za naprednim metodama za praćenje i predviđanje njihovog stanja postaje sve izraženija. Ovo istraživanje fokusira se na primjenu tehnika mašinskog učenja za precizno predviđanje stanja zdravlja litijum-jonskih baterija.

## 1.1. Motivacija

Litijum-jonske baterije igraju ključnu ulogu u savremenom svijetu, služeći kao osnovna komponenta u nizu elektronskih uređaja, od mobilnih telefona i laptopova do električnih vozila i sistema za skladištenje obnovljive energije. Prednosti litijum-jonskih baterija proizlaze iz visoke energetske efikasnosti, dugog životnog vijeka i relativno niskog stepena samopražnjenja u poređenju sa starim tehnologijama baterija, poput nikl-kadmijumskih (NiCd) i nikl-metal-hidridnih (NiMH) baterija. Međutim, s obzirom na njihovu široku primjenu i sve veći oslonac društva na elektronske uređaje, upravljanje njihovim performansama tokom vremena postalo je izazov.

Degradacija litijum-jonskih baterija je proces koji se ne može izbjeći; ona se manifestuje kao smanjenje kapaciteta i povećanje unutrašnje otpornosti, što direktno utiče na vrijeme rada uređaja koji koriste ove baterije. Na primjer, baterija u električnom vozilu može izgubiti dio svog kapaciteta tokom vremena, što rezultuje smanjenjem dometa vozila između punjenja. Ovaj problem je posebno izražen u primjenama gdje je predvidivo i pouzdano ponašanje baterija od ključnog značaja, kao što su električna vozila i kritični sistemi napajanja.

Pored tehničkih izazova, degradacija baterija ima i ozbiljne ekonomske implikacije, kao što je povećanje troškova održavanja i zamjene baterija [1]. Na primjer, kompanije koje posjeduju velik broj električnih vozila moraju uzeti u obzir troškove zamjene baterija, što može znatno povećati ukupne troškove vlasništva. Takođe, u sektoru obnovljivih izvora energije, degradacija baterija koje se koriste za skladištenje energije može smanjiti efikasnost ovih sistema, što dovodi do povećanja operativnih troškova i smanjenja isplativosti investicija [2].

U svjetlu ovih izazova, motivacija za ovo istraživanje je višestruka. S jedne strane, precizno predviđanje stanja zdravlja litijum-jonskih baterija može omogućiti optimizovano upravljanje njihovim životnim ciklusom, čime se produžava njihov radni vijek i smanjuju

troškovi. S druge strane, razvoj novih metoda za predviđanje stanja zdravlja može doprinjeti unapređenju tehnologije baterija i otvoriti nove mogućnosti za njihovu primjenu u različitim industrijama. Tačno predviđanje stanja zdravlja omogućava proaktivno održavanje i smanjenje rizika od neočekivanih kvarova, čime se poboljšava pouzdanost sistema koji zavise od litijum-jonskih baterija [3].

## 1.2. Postojeći radovi

Istraživanja u oblasti predikcije stanja zdravlja litijum-jonskih baterija su raznolika i obuhvataju širok spektar tradicionalnih pristupa, uključujući fizičke, hemijske i modelom bazirane metode. Tradicionalne fizičke metode za procjenu stanja zdravlja uključuju mjerenje unutrašnje otpornosti, kapaciteta i Kulonove efikasnosti baterija[1], pri čemu se ovi parametri koriste kao indikatori stanja zdravlja baterije. Na primjer, povećanje unutrašnje otpornosti tokom vremena je često povezano sa povećanjem otpora elektrolita i pasivnog sloja na elektrodama, što smanjuje efikasnost punjenja i pražnjenja baterije [4]. Iako ove metode pružaju vrijedne uvide, one su često nedovoljne za tačnu i pravovremenu predikciju stanja zdravlja baterije, posebno u kompleksnim aplikacijama gdje su mnogobrojni faktori uticaja [5].

Paralelno sa fizičkim pristupima, hemijski modeli su takođe korišteni za analizu degradacije baterija. Ovi modeli često uključuju detaljne simulacije hemijskih reakcija unutar baterije, kao što su prelazi litijuma kroz elektrolit i formiranje SEI (eng. *solid-electrolyte interphase*) sloja na anodi [6]. Ove simulacije omogućavaju istraživačima da prouče uticaj različitih faktora, kao što su temperatura, ciklusi punjenja i pražnjenja, na dugotrajnu stabilnost baterije. Međutim, hemijski modeli su složeni i zahtjevaju značajne računske resurse, što ograničava njihovu primjenu u realnim vremenskim sistemima [7].

U posljednje vrijeme, sa razvojem mašinskog učenja, došlo je do značajnog napretka u primjeni ovih tehnologija za predikciju stanja zdravlja. Neuronske mreže, uključujući duboke neuronske mreže (eng. *deep neural network*; DNN), konvolucione neuronske mreže (eng. *convolutional neural network*; CNN), i rekurentne neuronske mreže, specifično duge kratkoročne memorije (eng. *long short-term memory*; LSTM), pokazale su se kao veoma efikasne u prepoznavanju obrazaca u podacima o ciklusima baterija i predikciji njihovog stanja zdravlja [3, 8]. Studija autora Wang i kolega koristi CNN za analizu vremenskih serija podataka o napajanju i pražnjenju baterija, pokazujući da ova metoda može značajno poboljšati tačnost predikcije SOH-a u poređenju sa tradicionalnim metodama [2].

Pored neuronskih mreža, istraživači su takođe ispitali primjenu drugih algoritama mašinskog učenja, kao što su slučajna šuma (eng. *random forest*; RF), regresija pomoću potpornih vektora (eng. *support vector regression*; SVR) i regresija pomoću Gausovih procesa (eng. *Gaussian process regression*; GPR). Istraživanje Lu i saradnika pokazalo je da kombinacija RF-a i GPR-a može postići visoku tačnost u predikciji stanja zdravlja, posebno kada se koriste veliki skupovi podataka sa detaljnim informacijama o ciklusima baterije [2, 9].

[1] Kulonova efikasnost baterije, poznata i kao *Coulombic efficiency*, predstavlja odnos između količine naelektrisanja (izraženog u amper-satima) koji se potroši iz baterije tokom pražnjenja i količine naelektrisanja koji se unese u bateriju tokom punjenja. Izražava se kao postotak i služi kao mjera efikasnosti baterije.

Ove tehnike su posebno korisne kada se radi sa nelinearnim i visokodimenzionalnim podacima, kao što su oni koji se dobijaju iz operativnih ciklusa litijum-jonskih baterija.

Neki istraživači su se okrenuli korištenju inovativnih tehnika kako bi unaprijedili procjenu stanja zdravlja litijum-jonskih baterija. Jedna od tih tehnika je dekompozicija varijacionih modova (eng. *variational mode decomposition*; VMD), koja omogućava precizno izdvajanje relevantnih signala iz kompleksnih podataka o baterijama. Ova metoda je posebno korisna u analizi nelinearnih i nestacionarnih vremenskih serija, što je čest slučaj kod podataka o baterijama. Kombinovanjem ove tehnike sa optimizacijom regresije pomoću potpornih vektora (SVR), istraživači su uspjeli da poboljšaju tačnost modela za procjenu stanja zdravlja baterija. SVR, kao moćan alat za regresione zadatke, omogućava modelima da prepoznaju složene odnose u podacima, čime se povećava preciznost predikcija [10].

Pored toga, drugi istraživači su se fokusirali na korištenje prenosnog učenja i hibridnih modela kako bi poboljšali sposobnost generalizacije algoritama mašinskog učenja. Prenosno učenje omogućava modelima da iskoriste znanje stečeno na jednom skupu podataka i primjene ga na slične zadatke sa drugim skupovima podataka. Ova tehnika je posebno korisna kada su dostupni podaci ograničeni ili kada se model suočava sa novim situacijama koje nisu bile prisutne u treniranju. Kombinovanjem prenosnog učenja sa hibridnim modelima (modeli nastali kombinacijom različitih algoritama ili tehnika u mašinskom učenju koja omogućava bolje rezultate nego što bi se postiglo korištenjem pojedinačnih modela), istraživači su uspjeli da postignu visoku tačnost i pouzdanost predikcija, čak i u složenim scenarijima [11].

Da bi se dalje procijenila efikasnost različitih algoritama, sprovedene su i komparativne studije koje su analizirale performanse različitih metoda. U ovim studijama, algoritmi kao što su *XGBoost*, SVR i RF pokazali su svoje prednosti i slabosti u kontekstu predikcije stanja zdravlja baterija. *XGBoost*, poznat po svojoj sposobnosti da brzo i efikasno obradi velike skupove podataka, čest je bio izbor za zadatke regresije. S druge strane, SVR je pokazao svoju snagu u prepoznavanju složenih obrazaca u podacima, dok je RF, sa svojom sposobnosti da kombinuje rezultate više stabala odluke, pružio robusna rješenja koja su otporna na pretjeranu optimizaciju modela [12-14].

Osim toga, detaljno su ispitane i arhitekture neuronskih mreža, kao što su LSTM i CNN. LSTM mreže su se pokazale posebno efikasnim u radu sa vremenskim serijama, što ih čini pogodnim za analizu podataka o ciklusima punjenja i pražnjenja baterija. Sa druge strane, CNN arhitekture su se istakle u prepoznavanju obrazaca u podacima kroz automatsko izdvajanje relevantnih karakteristika, što je omogućilo unapređenje tačnosti predikcija stanja zdravlja baterija [15].

Takođe, istraživači su ispitali efikasnost mašinskih algoritama kao što su *Ridge* i *Lasso* u procjeni stanja zdravlja baterija. Jedno uobičajeno zapažanje u literaturi jeste potreba za dubljim razumjevanjem promjena u parametrima stanja baterija, uprkos poboljšanjima u efikasnosti koja su postignuta korištenjem metoda mašinskog učenja [12].

Jedan od glavnih izazova u primjeni mašinskog učenja za predikciju stanja zdravlja baterija je nedostatak standardizacije u načinu obrade podataka. Različiti istraživači često koriste različite metodologije za pripremu podataka, što dovodi do značajnih razlika u performansama algoritama koji su testirani na tim podacima. Na primjer, dok jedni istraživači

koriste jednostavne tehnike normalizacije i skaliranja podataka, drugi primjenjuju složene metode augmentacije podataka kako bi poboljšali generalizaciju modela. Zhang sa koautorima naglašava da nepostojanje jedinstvenih standarda za obradu podataka predstavlja prepreku za direktnu uporedivost rezultata različitih istraživanja i otežava izbor najefikasnijih tehnika za specifične primjene [16] .

Dodatni izazov je pretjerana optimizacija modela za specifične skupove podataka, što smanjuje njihovu sposobnost za generalizaciju na nove podatke. Ovaj fenomen poznat je kao *overfitting*, i jedan je od glavnih izazova u primjeni mašinskog učenja za predviđanje stanja zdravlja baterije. Na primjer, *Yin*-ova studija pokazuje da modeli koji postižu visoku tačnost na trening skupu podataka često pokazuju znatno lošije rezultate kada se testiraju na nepoznatim podacima, što ukazuje na potrebu za boljom generalizacijom modela [14].

U Tabeli 1.1 prikazani su ključni rezultati iz literature, zajedno sa odgovarajućim vrijednostima srednje kvadratne greške (eng. *mean squared error*; MSE) koje su povezane sa procjenom SOH-a. Vrijednosti MSE koje su označene zvjezdicom izračunate su korištenjem (1.1) na osnovu srednje apsolutne greške (MAE) prikazane u literaturi [17].

$$MSE = (1{,}25 \times MAE)^2 \qquad (1.1)$$

Varijacije u MSE vrijednostima koje su primjećene između različitih studija mogu se objasniti brojnim faktorima. Među njima su različiti skupovi podataka, tehnike pretprocesiranja, kao i razlike u arhitekturama neuronskih mreža, izboru hiperparametara i načinu na koji su podaci podeljeni između treninga i testiranja.

*Tabela 1.1. Poređenje rezultata iz literature*

| | Algoritam mašinskog učenja | MSE | | Reference |
|---|---|---|---|---|
| | | Minimalna vrijednost | Maksimalna vrijednost | |
| 1. | MLP NN | 0.0247 | x | [12] |
| 2. | CNN | 0.0017 | 0.0046 | [8] |
| **3.** | ***CatBoost*** | **0.0001** | **x** | **[14]** |
| **4.** | ***XGBoost*** | **0.0001** | **x** | **[18]** |
| | | **0.0001** | **0.0006** | **[14]** |
| 5. | *Random Forest* | 0.0009 | 0.0072 | [14] |
| 6. | *Lasso* | 0.0034* | x | [19] |
| 7. | *Ridge* | 0.0017* | x | [19] |
| 8. | *Linear Regression* | 0.0018* | x | [19] |
| 9. | LSTM NN | 0.0031 | 0.0055 | [3] |
| | | 0.0340 | 0.0410 | [15] |
| 10. | SVM | 0.0030 | 0.0160 | [15] |
| | | 0.0002 | 0.0010 | [18] |

Iako ovi rezultati predstavljaju korisne referentne tačke u ovom području, važno je imati na umu metodološke razlike između studija, kao što su upravljanje podacima, struktura mreža i podešavanje hiperparametara. Ove razlike zahtjevaju oprez pri tumačenju rezultata, jer oni nisu definitivni pokazatelji potencijala algoritma, već više ukazuju na performanse pod različitim uslovima. Ovo ukazuje na potrebu za standardizovanim metodama koje bi omogućile pouzdanije poređenje i dalji razvoj tehnika mašinskog učenja za konzistentnu i tačnu predikciju stanja zdravlja baterija.

Glavni cilj ovog istraživanja je poređenje rezultata različitih algoritama mašinskog učenja pod identičnim uslovima, kako bi se ovi izazovi riješili.

## 1.3. Problemi

Predikcija stanja zdravlja litijum-jonskih baterija predstavlja kompleksan izazov zbog složenih interakcija između hemijskih i fizičkih procesa koji se odvijaju unutar baterije tokom njenog životnog ciklusa. S obzirom na to da je degradacija baterije neizbježna, ali i teško predvidiva, ključni problem leži u razvoju modela koji mogu tačno predvidjeti stanje zdravlja na osnovu dostupnih podataka o ciklusima punjenja, pražnjenja i impedanse.

Jedan od najvećih problema u ovom domenu je nedostatak standardizacije u načinu na koji se podaci prikupljaju, obrađuju i analiziraju. Različiti istraživači često koriste različite metodologije za pripremu podataka, što otežava direktnu uporedivost rezultata između različitih studija. Ovo može dovesti do situacije gdje jedan algoritam pokazuje superiorne performanse na jednom skupu podataka, dok isti taj algoritam ne uspijeva da postigne zadovoljavajuće rezultate na drugom skupu zbog različitih metoda obrade podataka. Ovaj nedostatak standardizacije predstavlja značajnu prepreku za razvoj univerzalnih modela koji bi se mogli primjenjivati u različitim kontekstima i aplikacijama.

Još jedan značajan problem je pretjerana optimizacija modela za specifične skupove podataka, što dovodi do loše generalizacije na nove podatke. U kontekstu predikcije stanja zdravlja baterije, ovo može značiti da model neće biti pouzdan u stvarnim operativnim uslovima, gdje se podaci razlikuju od onih na kojima je model treniran.

## 1.4. Doprinosi

Ovo istraživanje donosi nekoliko ključnih doprinosa. Na osnovu analize postojećih arhitektura neuronskih mreža i algoritama mašinskog učenja, prilagođene su i optimizovane arhitekture koje su najpogodnije za predikciju zdravlja litijum-jonskih baterija. Evaluacija je izvršena korištenjem relevantnih metrika kao što su MSE, $R^2$ kod regresije i tačnost i F1 metrika kod klasifikacije čime su obučeni modeli postigli visoku pouzdanost i primjenljivost.

Korištenjem uniformnih metodologija za obradu podataka omogućena je direktna komparacija različitih modela i pristupa. Dodatno, kreirani su obimni i raznovrsni skupovi podataka kroz augmentaciju, čime je poboljšana generalizacija modela u različitim kontekstima.

Predložena arhitektura je validirana na realnim podacima, a rezultati su evaluirani prema industrijskim standardima, potvrđujući veoma dobre performanse u poređenju sa postojećim metodama. Dio rezultata je objavljen u [20], gdje je obuhvaćen sav rad iz ovog master rada vezan za eksperimente sa NASA skupom podataka, dodatno potvrđujući značaj i relevantnost ovog istraživanja. S obzirom da su eksperimenti djeljeni između tog naučnog rada i ovog master rada, dosta sadržaja je iskorišteno u ovom radu.

Cijela implementacija, uključujući skripte za optimizaciju modela, obradu podataka, i izvršene eksperimente, dostupna je na *GitHub* platformi, omogućavajući transparentnost i reproduktivnost rezultata[2][3].

## 1.5. Pregled sadržaja

Ovaj rad je strukturisan u šest glavnih glava. Prva glava je uvod, koji daje pregled motivacije, problema, ciljeva i doprinosa istraživanja. U drugoj glavi se razmatraju osnovne metrike koje se koriste za procjenu zdravlja baterija, kao i faktori koji utiču na degradaciju performansi baterije tokom vremena. U trećoj glavi su opisane statističke metrike kojima se ocjenjuju performanse modela mašinskog učenja i korišteni algoritmi mašinskog učenja i neuronskih mreža, uključujući njihove prednosti i slabosti u kontekstu predikcije zdravlja baterija. U četvrtoj glavi su opisani korišteni skupovi podataka, gdje se detaljno analiziraju podaci korišteni za treniranje i validaciju modela. U glavi pet su izloženi eksperimentalni rezultati koji pružaju uvid u performanse različitih modela i algoritama, kao i njihovu evaluaciju na osnovu prethodno definisanih metrika. Glava šest predstavlja zaključak, koji sumira ključne nalaze istraživanja, diskutuje o implikacijama rezultata i predlaže pravce za buduće istraživanje.

[2] https://github.com/stepanov1997/battery_health_prediction_using_ml_master_thesis
[3] Napomena: svaki poseban eksperiment se nalazi u različitoj git grani.

# 2. Pojam zdravlja litijum-jonske baterije

## 2.1. Uvod i istorijski pregled

Litijum-jonske baterije, razvijene tokom 1970-ih i 1980-ih, izvršile su revoluciju u industriji prenosivih elektronskih uređaja i danas su osnovna tehnologija za skladištenje energije u mnogim sektorima, uključujući električna vozila, mobilne uređaje i sisteme za skladištenje obnovljive energije [21]. Prva komercijalna litijum-jonska baterija uvedena je na tržište od strane *Sony Corporation* [4] 1991. godine, što je označilo početak ere visokih energetskih gustina, sa vrijednostima do 150-200 Wh/kg i dugog životnog vijeka baterija, koji može iznositi preko 1000 ciklusa punjenja i pražnjenja [22]. Njihova upotreba u električnim vozilima, od strane kompanija kao što su Tesla Motors[5], Nissan[6] i General Motors[7], samo je povećala njihovu važnost, stavljajući akcenat na potrebu za preciznim praćenjem stanja zdravlja baterija [23].

U početnim fazama, SOH baterija se procjenjivao jednostavnim metodama kao što su praćenje kapaciteta i unutrašnje otpornosti tokom vremena. Istraživanja iz ranih 2000-ih pokazala su da se povećanje unutrašnjeg otpora može povezati sa formiranjem SEI sloja, što direktno utiče na smanjenje kapaciteta baterije [24]. Ova metoda je bila dovoljno efikasna za rane primjene litijum-jonskih baterija, ali nije bila dovoljno precizna za moderne primjene koje zahtijevaju dugoročnu pouzdanost, kao što su električna vozila i sistemi za skladištenje energije.

Kako su potrebe za preciznosti procjene SOH-a rasle, istraživači su počeli da razvijaju složenije modele koji uzimaju u obzir više parametara, uključujući Kulonovu efikasnost, temperaturu, i promjene u hemijskom sastavu baterije [25]. Ovi modeli su često zasnovani na analizi podataka iz velikog broja ciklusa punjenja i pražnjenja, omogućavajući identifikaciju obrazaca koji su teško uočljivi pomoću jednostavnih empirijskih metoda [26].

Posljednjih godina, tehnike mašinskog učenja su postale ključni alat u analizi i predikciji SOH-a. Ovi pristupi omogućavaju analizu velikih skupova podataka i identifikaciju složenih obrazaca degradacije koji su prisutni u savremenim litijum-jonskim baterijama [27]. Na primjer, korištenje dubokih neuronskih mreža za analizu vremenskih serija podataka o ciklusima punjenja i pražnjenja pokazalo je značajan napredak u tačnosti predikcija [28].

## 2.2. SOH metrika

SOH metrika je osnovna metrika koja kvantifikuje trenutni status baterije u poređenju sa njenim nominalnim stanjem kada je bila nova. Ova metrika je ključna za razumijevanje koliko je baterija degradirala tokom svog životnog ciklusa, i obično se izražava kao procenat trenutnog kapaciteta u odnosu na nominalni kapacitet baterije. Tradicionalni pristupi za

[4] https://www.sony.net/
[5] https://www.tesla.com/
[6] https://www.nissanusa.com/
[7] https://www.gm.com/

izračunavanje SOH-a oslanjali su se na osnovne parametre kao što su kapacitet i unutrašnja otpornost baterije.

Kapacitet baterije predstavlja količinu električne energije koju baterija može pohraniti i isporučiti pod određenim uslovima, obično izražen u miliamper-satima (mAh) ili amper-satima (Ah). To je ključna karakteristika baterije koja određuje koliko dugo može napajati uređaj pre nego što se isprazni. Kapacitet se s vremenom smanjuje zbog procesa starenja i upotrebe, što utiče na ukupne performanse i trajanje rada baterije.

Jedan od najjednostavnijih modela za procjenu SOH-a koristi odnos trenutnog i nominalnog kapaciteta baterije, pri čemu je SOH definisan u (2.1).

$$SOH = \frac{C_{trenutni}}{C_{nominalni}} \times 100\% \quad (2.1)$$

Iako je ova metoda jednostavna i široko primijenjena, ona ima svoja ograničenja, jer ne uzima u obzir faktore kao što su promjena unutrašnjeg otpora i uticaj temperature, što može značajno uticati na performanse baterije [29].

Drugi pristup je praćenje promjene unutrašnjeg otpora baterije, koji može biti indikativan za stanje zdravlja baterije, jer se unutrašnji otpor obično povećava kako baterija stari. Promjene u unutrašnjem otporu često su povezane sa formiranjem SEI sloja, degradacijom elektrolita, i mehaničkim promjenama unutar baterijskih ćelija [30]. Studije su pokazale da se unutrašnji otpor može efikasno koristiti za predikciju SOH-a, posebno u kombinaciji sa drugim parametrima kao što su temperatura i Kolumbova efikasnost [31].

Savremeni modeli za procjenu SOH-a koriste složenije algoritme, često integrisane sa metodama mašinskog učenja. Na primjer, Zhang i saradnici su razvili model koji koristi CNN-ove za analizu vremenskih serija podataka, omogućavajući precizniju procjenu SOH-a u realnom vremenu [26]. Ovaj pristup ne samo da uzima u obzir više parametara odjednom, već omogućava i adaptivnu procjenu, gdje se model prilagođava promjenama u uslovima rada baterije.

Jedan od inovativnih pristupa u procjeni SOH-a uključuje korišćenje hibridnih modela koji kombinuju fizičke modele sa mašinskim učenjem. Na primjer, hibridni model, razvijen od strane Lu-a i saradnika, integriše hemijske simulacije sa metodama dubokog učenja, omogućavajući precizniju procjenu SOH-a čak i u složenim operativnim uslovima [32]. Ovaj pristup pokazao je značajna poboljšanja u tačnosti predikcija u poređenju sa tradicionalnim metodama, ali i dalje zahtijeva visoke računske resurse.

U ovom radu SOH metrika se pokazala kao metrika koja je dovoljno dobra, pogotovo kad se mašinskim učenjem uspostavi zavisnost svih važnih karakteristika baterije sa SOH-om.

## 2.3. Metrika preostalog vijeka trajanja baterije

Pored SOH-a, metrika preostalog vijeka trajanja baterije (eng. *remaining useful life*; RUL) je ključna za predikciju koliko dugo baterija može nastaviti da funkcioniše prije nego što postane neupotrebljiva. Tradicionalno, RUL se procjenjivao na osnovu linearne ekstrapolacije

trenutnih stopa degradacije, pri čemu se analiziraju istorijski podaci o ciklusima punjenja i pražnjenja [33].

Međutim, linearni modeli su često nedovoljni za složene primjene gdje degradacija nije linearna. Na primjer, promjene u temperaturi, stopa pražnjenja i način korištenja mogu značajno ubrzati ili usporiti degradaciju baterije, što linearni modeli ne uzimaju u obzir [26]. Zbog toga su razvijeni napredniji modeli koji koriste mašinsko učenje za predikciju RUL-a.

Jedan od najčešće korištenih algoritama za predikciju RUL-a su rekurentne memorijske mreže (LSTM), koje su posebno efikasne u analizi vremenskih serija. Na primjer, Li sa saradnicima su pokazali da LSTM mreže mogu značajno poboljšati tačnost predikcija RUL-a u poređenju sa tradicionalnim metodama, jer su sposobne da modeliraju nelinearne obrasce degradacije. Ovaj pristup omogućava precizniju procjenu preostalog vijeka trajanja baterije, uzimajući u obzir varijabilnost u uslovima rada [34].

Probabilistički modeli, kao što su *Bayesian Networks* i *Monte Carlo* simulacije, koriste se za procjenu neizvijesnosti u predikciji RUL-a. Ovi modeli omogućavaju kreiranje distribucije mogućih vrijednosti RUL-a, umjesto jedne determinističke vrijednosti, što pruža detaljnije informacije o vjerovatnim scenarijima degradacije [35].

Pan i kolege su razvili metod za predikciju RUL-a litijum-jonskih baterija, koji uzima u obzir regeneraciju kapaciteta i nasumične fluktuacije. Ovaj metod koristi empirijsku dekompoziciju modova (EMD) za razdvajanje dugoročnog trenda degradacije, dok se za predikciju budućeg trenda koristi LSTM memorijska mreža. Pored toga, GPR se koristi za predikciju regeneracije kapaciteta, dok se nasumične fluktuacije modeliraju pomoću stabilne distribucije [36].

Upotreba naprednih modela za procjenu RUL-a, uključujući probabilističke i pristupe zasnovane na mašinskom učenju, predstavlja značajan napredak u pouzdanosti i preciznosti predikcija, omogućavajući efikasnije planiranje održavanja i optimizaciju korištenja baterija u raznim industrijama.

# 3. Opis korištenih algoritama

U ovom poglavlju obrađene su ključne komponente metodološkog pristupa korištenog u ovoj studiji za predikciju zdravlja litijum-jonskih baterija. Precizna predikcija stanja zdravlja (SOH) baterija od suštinske je važnosti za optimizaciju performansi, produženje životnog vijeka i održavanje sigurnosti ovih uređaja. Da bi se postigli ovi ciljevi, neophodno je koristiti adekvatne metrike za procjenu kvaliteta predikcija, kao i napredne algoritme mašinskog i dubokog učenja koji mogu efikasno obraditi složene i velike skupove podataka.

Prvi dio ovog poglavlja fokusira se na metrike koje su korištene za evaluaciju modela, pružajući osnovu za razumijevanje kako se mjeri tačnost i pouzdanost predikcija. Kroz pažljivu analizu ovih metrika, moguće je steći uvid u efikasnost modela i identifikovati oblasti za dalje unapređenje.

Nakon toga, detaljno su opisani algoritmi koji su primijenjeni u ovoj studiji. Ovi algoritmi, bilo da su bazirani na regresiji ili klasifikaciji, omogućavaju modelima da izvuku vrijedne uvide iz podataka i donesu precizne prognoze o stanju zdravlja baterija. Svaki algoritam je opisan kroz svoju strukturu, način funkcionisanja, te specifične prednosti u kontekstu predikcije SOH-a.

## 3.1. Regresivne metrike

Regresija je statistička metoda koja se koristi za modelovanje odnosa između zavisne promjenljive (koja se pokušava predvidjeti) i jedne ili više nezavisnih promjenljivih (koje se koriste kao ulazni podaci). U kontekstu mašinskog učenja, regresija se koristi za predviđanje kontinualnih vrijednosti. Na primjer, moguće je koristiti regresiju da bi se predvidjela preostala upotrebljivost litijum-jonske baterije na osnovu različitih faktora poput broja ciklusa punjenja, temperature rada i kapaciteta baterije.

U regresiji se pravi model koji nastoji uspostaviti preslikavanje između nezavisnih i zavisnih promjenljivih. Cilj je minimizovati razliku između stvarnih vrijednosti i predviđenih vrijednosti koje daje model. Ova razlika se često naziva greška.

### 3.1.1. Srednja kvadratna greška

Srednja kvadratna greška je jedna od najčešće korištenih metrika za procjenu kvaliteta regresionih modela. MSE mjeri prosječnu kvadratnu razliku između stvarnih vrijednosti i vrijednosti koje je model predvidio. Računa se tako što se suma kvadrata razlika između stvarnih i predviđenih vrijednosti podijele sa ukupnim brojem tih vrijednosti. Način računanja MSE metrike je prikazana u (3.1), gdje je $y_i$ stvarna vrijednost, $\widehat{y_i}$ predviđena vrijednost, a $n$ broj podataka. Niža vrijednost MSE metrike ukazuje na veću preciznost modela.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \qquad (3.1)$$

### 3.1.2. Koeficijent determinacije (R-kvadrat)

Koeficijent determinacije ili R-kvadrat (eng. *R-squared*; $R^2$) je metrika koja pokazuje koliko dobro nezavisne varijable objašnjavaju varijabilnost zavisne varijable u regresionom modelu. R-kvadrat vrijednost se kreće između 0 i 1, gdje vrijednost bliža 1 znači da model bolje objašnjava varijabilnost podataka. $R^2$ se računa pomoću (3.2).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (3.2)$$

Promjenljiva $\bar{y}$ predstavlja prosječnu vrijednost stvarnih podataka. Kad je vrijednost $R^2$ bliža 1 znači da je model veoma dobar u predviđanju stvarnih podataka.

## 3.2. Klasifikacione metrike

Klasifikacija je metoda mašinskog učenja koja se koristi za razvrstavanje podataka u različite kategorije. Za razliku od regresije, koja predviđa kontinualne vrijednosti, klasifikacija se bavi diskretnim kategorijama ili klasama. Na primjer, klasifikacija se može koristiti za identifikaciju vrste e-maila kao "spam" ili "nije spam".

### 3.2.1. Matrica konfuzija

Matrica konfuzije je alat koji omogućava dublju analizu performansi klasifikacionog modela. Prikazuje koliko puta su stvarne instance svake klase klasifikovane u svaku od mogućih klasa. Matrica konfuzije ima oblik prikazan u Tabela 3.1.

*Tabela 3.1. Struktura matrice konfuzija*

| | **Predviđeno pozitivno** | **Predviđeno negativno** |
|---|---|---|
| **Stvarno pozitivno** | TP | FN |
| **Stvarno negativno** | FP | TN |

Gdje su:

- TP (eng. *true positive*) predstavlja broj ispravno predviđenih elemenata pozitivne klase.
- FP (eng. *false positive*) predstavlja broj neispravno predviđenih elemenata pozitivne klase.
- FN (eng. *false negative*) predstavlja broj neispravno predviđenih elemenata negativne klase.
- TN (eng. *true negative*) predstavlja broj ispravno predviđenih elemenata negativne klase.

### 3.2.2. Tačnost

Tačnost (eng. *accuracy*) je osnovna metrika za procjenu kvaliteta klasifikacionih modela. Izračunava se kao odnos broja ispravno klasifikovanih instanci i ukupnog broja instanci. U (3.3) je prikazano kako se računa tačnost.

$$\text{Tačnost} = \frac{\text{Broj ispravno klasifikovanih instanci}}{\text{Ukupan broj instanci}} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.3)$$

Na primjer, ako postoji model koji ispravno klasifikuje 90 od 100 podataka, tačnost bi bila 90%. Iako je tačnost korisna metrika, ona ne daje potpunu sliku kada su podaci neuravnoteženi, tj. kada postoji mnogo više instanci jedne klase u odnosu na drugu.

### 3.2.3. Preciznost

Preciznost je metrika koja mjeri tačnost modela u prepoznavanju pozitivnih instanci. Preciznost pokazuje koliki je udio ispravno predviđenih pozitivnih slučajeva u odnosu na sve slučajeve koje je model označio kao pozitivne. Drugim riječima, preciznost odgovara na pitanje: "Od svih instanci koje je model označio kao pozitivne, koliko njih je zaista pozitivno?". U (3.4) je prikazano kako se računa preciznost.

$$\text{Preciznost} = \frac{TP}{TP+FP} \quad (3.4)$$

Visoka preciznost znači da model pravi malo grešaka kada označava instancu kao pozitivnu, tj. ima malo lažnih pozitivnih (FP). Međutim, visoka preciznost ne garantuje da će model prepoznati sve pozitivne instance, što dovodi do sljedeće metrike - odziva.

### 3.2.4. Odziv

Odziv je metrika koja mjeri sposobnost modela da prepozna sve pozitivne instance. Odziv odgovara na pitanje: "Koliko od stvarnih pozitivnih instanci je model uspio ispravno prepoznati?" Drugim riječima, odziv pokazuje koliki je procenat ispravno predviđenih pozitivnih slučajeva u odnosu na ukupan broj stvarno pozitivnih slučajeva. U (3.5) je prikazano kako se računa odziv.

$$\text{Odziv} = \frac{TP}{TP+FN} \quad (3.5)$$

Visok odziv znači da model prepoznaje većinu stvarno pozitivnih instanci, ali to ne mora značiti da je model precizan u predviđanjima, jer može biti mnogo lažnih pozitivnih slučajeva. Preciznost i odziv su često u sukobu - povećanje preciznosti može smanjiti odziv i obrnuto. F1 metrika se koristi kao harmonijska sredina preciznosti i odziva, kako bi se postigao balans između ovih metrika i sveo njihov informativni sadržaj na jedan broj u klasifikacionim modelima.

### 3.2.5. F1 metrika

F1 metrika predstavlja harmonijsku sredina između preciznosti (eng. *precision*) i odziva (eng. *recall*). Koristi se u slučaju neuravnoteženih klasa, jer uzima u obzir kako tačnost tako i pokrivenost modela. Preciznost mjeri koliko su predikcije tačne kada model klasifikuje primjer kao pozitivnu klasu, dok odziv mjeri koliko je pozitivnih klasa tačno identifikovano od ukupnog broja stvarnih pozitivnih klasa. U (3.6) je navedeno kako se računa F1 metrika.

$$F1 = 2 \times \frac{\text{Preciznost} \times \text{Odziv}}{\text{Preciznost} + \text{Odziv}} \qquad (3.6)$$

F1 metrika je posebno korisna u situacijama kada su pogrešne pozitivne i pogrešne negativne klasifikacije jednako važne.

### 3.2.6. Primjer matrice konfuzije i analiza

Pretpostavimo da postoji model koji klasifikuje poruke elektronske pošte kao "spam" ili "nije spam". Nakon testiranja na 100 poruka, matrica konfuzije izgleda ovako:

| | **Predviđeno "spam"** | **Predviđeno "nije spam"** |
|---|---|---|
| **Stvarno "spam"** | 40 | 10 |
| **Stvarno "nije spam"** | 5 | 45 |

U ovom primjeru:

- TP = 40 (40 e-mailova su ispravno klasifikovana kao spam).
- FP = 5 (5 e-mailova su netačno klasifikovana kao spam).
- FN = 10 (10 e-mailova su netačno klasifikovana kao nije spam).
- TN = 45 (45 e-mailova su ispravno klasifikovana kao nije spam).

Iz ove matrice možemo izračunati različite metrike:

- Tačnost: $\frac{40 + 45}{100} = 0.85$ (ili 85%).
- Preciznost: $\frac{40}{40 + 5} = 0.89$ (ili 89%).
- Odziv: $\frac{40}{40 + 10} = 0.8$ (ili 80%).
- F1 metrika: $2 \times \frac{0.89 \times 0.8}{0.89 + 0.8} = 0.84$ (ili 84%).

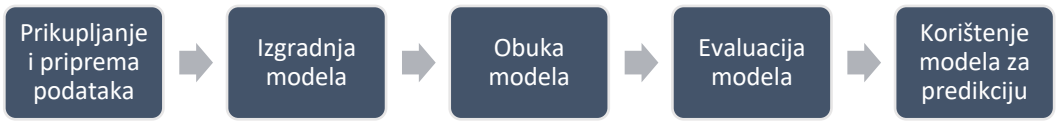
Ove metrike omogućavaju dublje razumijevanje kako model funkcioniše, što je korisno za donošenje odluka o njegovom poboljšanju ili prilagođavanju.

## 3.3. Klasični algoritmi mašinskog učenja

Klasični algoritmi mašinskog učenja predstavljaju osnovne alate za analizu podataka i donošenje odluka, oslanjajući se na matematičke modele koji izvode zaključke iz podataka na osnovu statističkih zakonitosti. Ovi algoritmi, kao što su regresija, klasifikacija, i metode

klasterizacije, koriste jednostavne, ali efikasne pristupe za modelovanje odnosa među promjenljivima. Klasični algoritmi su poznati po svojoj interpretabilnosti i relativno niskim zahtjevima za procesorskim resursima, što ih čini pogodnim za rješavanje mnogih inženjerskih problema. Međutim, u složenijim scenarijima, gdje osnovni modeli možda ne postižu zadovoljavajuće performanse, na scenu stupaju *gradient boosting* algoritmi. *Gradient boosting* algoritmi, kao što su *XGBoost* i *CatBoost*, poboljšavaju performanse kombinovanjem više jednostavnih modela u snažan ansambl. Ovi ansambl modeli iterativno treniraju jednostavne modele, fokusirajući se na ispravku grešaka iz prethodnih iteracija, čime postižu visoku tačnost i otpornost na *overfitting*, što ih čini ključnim alatima u modernim inženjerskim primjenama mašinskog učenja. U ovom poglavlju su opisani klasični algoritmi mašinskog učenja koji su korišteni u eksperimentima i kakva bi im mogla biti primjena kad je u pitanju određivanje SOH-a litijum-jonskih baterija.

Na Slici 3.1 su prikazani svi koraci u procesu mašinskog učenja koje dijele svi algorimi koji su navedeni u ovom poglavlju.




*Slika 3.1. Dijagram toka procesa mašinskog učenja*

Proces mašinskog učenja započinje prikupljanjem i pripremom podataka, gdje se sakupljaju relevantni podaci, čiste i pretprocesiraju kako bi bili spremni za analizu. Nakon toga, slijedi faza izgradnje modela, u kojoj se bira odgovarajući algoritam i definišu parametri modela, što omogućava modelu da uči iz podataka. Ključna komponenta ovog procesa je obuka modela, gdje se model trenira koristeći trening skup podataka kako bi optimizovao svoje parametre u cilju minimizacije grešaka u predviđanjima. U ovoj fazi svaki algoritam koristi specifičan proračun funkcije cijene, koja mjeri razliku između stvarnih i predviđenih vrijednosti, i to često određuje sam algoritam i njegovu primjenu. Nakon obuke, model se evaluira na testnom skupu podataka kako bi se procijenile tačnost i performanse, koristeći metričke performanse koje pokazuju koliko je model uspješan u predviđanju. Adekvatno obučen i evaluiran model koristi se za predikciju novih vrednosti na osnovu novih ulaznih podataka, omogućavajući donošenje odluka i predviđanja u stvarnim aplikacijama, što završava ciklus od obrade podataka do praktične primjene modela.

Važno je napomenuti i proces regularizacije. Regularizacija je tehnika koja se koristi u mašinskom učenju kako bi se spriječio *overfitting*. *Overfitting* nastaje kada model postane previše složen, prilagođavajući se svim detaljima i šumovima u trening podacima, što rezultuje loše performanse na novim, nepoznatim podacima. Regularizacija dodaje kaznene slobodne članove u funkciju cijene modela, čime se smanjuje složenost modela i podstiču jednostavnija rješenja koja bolje generalizuju na nove podatke.

Regularizacija je korisna jer balansira između tačnosti modela na trening podacima i njegove sposobnosti generalizacije na nove podatke. Primjenom regularizacije, model postaje robusniji i manje osjetljiv na varijabilnost u podacima, čime se poboljšavaju performanse na testnim i stvarnim podacima. Na primjer, u problemima gdje postoji veliki broj značajki, L1

regularizacija može biti korisna za identifikaciju i odabir najvažnijih obilježja, dok je L2 regularizacija korisna za stabilizaciju težina i smanjenje osjetljivosti modela na šum.

### 3.3.1. Metoda potpornih vektora

Metoda potpornih vektora (eng. *support vector networks*; SVM) je nadgledani algoritam mašinskog učenja koji se koristi kako za klasifikaciju, tako i za regresiju. SVM funkcioniše tako što pronalazi hiper-ravan u višedimenzionalnom prostoru koja na najbolji način razdvaja podatke različitih klasa. Osnovna ideja ovog algoritma je da se pronađe hiper-ravan koja maksimizuje razdvajanje klasa, odnosno da se maksimizuje margina između klasa. Marginu definišu potporni vektori, koji predstavljaju podskup skupa podataka za obučavanje najbližih hiper-ravni i ključni su za njeno definisanje [37].

SVM može koristiti različite jezgrene funkcije (eng. *kernel functions*) za transformaciju podataka u višedimenzionalni prostor u kojem su klase linearno odvojive. Ove jezgrene funkcije omogućavaju SVM-u da efikasno radi i sa podacima koji nisu linearno odvojivi u izvornom prostoru. Neke od najčešće korišćenih jezgrenih funkcija uključuju:

- Linearna jezgrena funkcija (koristi se kada su podaci već linearno odvojivi),
- Polinomska jezgrena funkcija (omogućava modelu da uči nelinearne odnose među podacima),
- Radijalna bazna funkcija (RBF; posebno pogodna za slučajeve gdje su klase odvojive nelinearno),
- Sigmoidalna jezgrena funkcija (često korištena kao alternativa za neuronske mreže).

Izbor jezgrene funkcije zavisi od prirode podataka i specifičnog problema koji se riješava. Transformacijom podataka pomoću odgovarajuće jezgrene funkcije, SVM može pronaći optimalnu hiper-ravan koja razdvaja klase u novom, višedimenzionalnom prostoru [38].

SVM algoritam započinje izračunavanjem jezgrene matrice, koja sadrži sve parove unutrašnjih proizvoda između uzoraka u proširenom prostoru. Za jezgrenu funkciju $K(x_i, x_j)$, matrica $K$ je definisana u (3.7), gdje $\phi(x)$ označava funkciju preslikavanja (eng. *feature mapping*) koja transformiše ulazni vektor $x$ u viši dimenzionalni prostor. Ovaj pristup omogućava SVM-u da pronađe optimalno odvajanje između klasa čak i u slučajevima kada podaci nisu linearno odvojivi u originalnom prostoru.

$$K_{ij} = \phi(x_i) \cdot \phi(x_j) \tag{3.7}$$

SVM koristi kvadratno programiranje (eng. *quadratic programming*) za maksimizaciju margine između klasa, pri čemu se poštuju ograničenja koja postavljaju potporni vektori. Funkcija cijene koja se maksimizuje je prikazana u (3.8). Parametri $w$ i $b$ predstavljaju parametre hiper-ravni, a $y_i$ klase koje odgovaraju trening uzorcima $x_i$.

$$\text{Funkcija cijene: } \frac{1}{2}\left|\left|w\right|\right|^2 \quad \text{uz ograničenje} \quad y_i(w \cdot \phi(x_i) + b) \geq 1, \quad \forall i \qquad (3.8)$$

Na osnovu potpornih vektora, SVM izračunava optimalnu hiper-ravan koja najbolje razdvaja klase, maksimizirajući marginu između njih. Kada je hiper-ravan jednom određena, koristi se za klasifikaciju novih podataka na osnovu njihovih pozicija u odnosu na tu ravan, omogućavajući precizno razdvajanje klasa i efikasnu klasifikaciju novih uzoraka.

SVM može efikasno raditi s visoko-dimenzionalnim podacima i pružiti tačne predikcije, što je korisno za klasifikaciju stanja baterija. Njegova otpornost na *overfitting* pomaže u održavanju preciznosti modela. S druge strane, SVM može biti računski intenzivan i zahtijevati puno memorije, naročito kod velikih skupova podataka. Podešavanje odgovarajućih parametara i izbora jezgra može biti složeno i zahtijeva pažljivo podešavanje kako bi se postigli optimalni rezultati.

### 3.3.2. Linearna regresija

Linearna regresija (eng. *linear regression*) je osnovni algoritam mašinskog učenja koji se koristi za modeliranje odnosa između zavisne promjenljive i jedne ili više nezavisnih promjenljivih. Cilj linearne regresije je pronaći linearni model (ravan ili hiperravan, ili prava linija u slučaju jedne nezavisne promjenljive) koji najbolje opisuje odnos između promjenljivih. Ovaj model se izražava sa (3.9), gdje su β koeficijenti regresije koji se procjenjuju metodom najmanjih kvadrata. Svaki $\beta_j$ predstavlja brzinu promjene (izvod $Y$ po $X_j$ ) zavisne promjenljive $Y$ u odnosu na promjenu nezavisne promjenljive $X_j$ , dok $\beta_0$ predstavlja vrijednost $Y$ kada su sve nezavisne promjenljive jednake nuli [39-45]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \qquad (3.9)$$

U linearnim regresionim modelima, specifičnosti uključuju definisanje zavisne promjenljive kao funkcije jedne ili više nezavisnih promjenljivih, gdje su koeficijenti $\beta_j$ nepoznate konstante koju treba procijeniti. Procjena koeficijenata vrši se metodom najmanjih kvadrata, koja minimizuje sumu kvadrata razlika između stvarnih vrijednosti $Y_i$ i predikcija $\widehat{Y}_i$. Funkcija cijene za linearne regresione modele može se predstaviti kao (3.10), gdje je $n$ broj posmatranja u trening skupu.

$$\text{Funkcija cijene: } \sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2 \qquad (3.10)$$

Linearna regresija je jednostavan i lako primjenljiv algoritam, što je čini pogodnom za brzo modelovanje i analizu podataka o litijum-jonskim baterijama. Njena prednost je u tome što omogućava lako razumijevanje kako različiti parametri baterije, poput napona, struje i temperature, utiču na njeno zdravlje, što olakšava interpretaciju rezultata i donošenje odluka o održavanju.

Međutim, linearna regresija ima i svoja ograničenja. U kontekstu baterija, možda neće biti tačna kada su odnosi između parametara i zdravlja baterije složeni i nelinearni, što je često slučaj. Takođe, osjetljiva je na ekstremne vrijednosti u podacima, što može uticati na tačnost predviđanja, pa može biti potrebna dodatna obrada podataka kako bi rezultati bili pouzdani.

### 3.3.3. *Ridge*

*Ridge* regresija je oblik regularizovane linearne regresije koji se koristi za rješavanje problema multikolinearnosti, tj. situacije kada su nezavisne promjenljive visoko međusobno korelisane. Dodavanjem L2 regularizacije ($\lambda \sum_{j=1}^{p} \beta_j^2$) funkciji cijene linearne regresije, *Ridge* regresija penalizuje velike koeficijente regresije, čime smanjuje varijansu modela i sprječava prekomjerno prilagođavanje (eng. *overfitting*). Ovaj pristup dodaje penalizacioni član koji je proporcionalan kvadratu veličine koeficijenata, čime se model "kažnjava" za prevelike vrijednosti koeficijenata [46].

U *Ridge* regresiji, koeficijenti $\beta_j$ se procjenjuju korištenjem optimizacije koja minimizuje penalizovana suma kvadrata razlika između stvarnih vrijednosti i predikcija. Ova optimizacija može biti prikazana kao funkcija cijene prikazane u (3.11), gdje je $\lambda$ parametar regularizacije, $\beta_j$ koeficijenti regresije, a $n$ broj uzoraka u trening skupu [47].

$$\text{Funkcija cijene: } \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad (3.11)$$

*Ridge* regresija je korisna za analizu podataka s visokom korelacijom među promjenljivima, jer efikasno smanjuje prekomjerno prilagođavanje modela, čime se poboljšava stabilnost i tačnost predikcija. Ovaj algoritam je posebno koristan kada se radi s velikim brojem promjenljivih, jer omogućava zadržavanje svih informacija bez značajnog gubitka tačnosti. Ipak, *Ridge* regresija nije uvijek prikladna za situacije gdje postoje nelinearni odnosi među podacima, jer njen linearni pristup može zanemariti kompleksne obrasce. Pored toga, budući da ne vrši selekciju promjenljivih, model može ostati opterećen suvišnim informacijama, što može otežati interpretaciju.

### 3.3.4. *Lasso*

*Lasso* regresija (eng. *least absolute shrinkage and selection operator*) je metoda regularizacije koja koristi L1 regularizaciju ($\lambda \sum_{j=1}^{p} |\beta_j|$) kako bi penalizovala apsolutne vrijednosti koeficijenata regresije. Za razliku od *Ridge* regresije, *Lasso* može rezultovati koeficijente čije su vrijednosti jednake nuli, što omogućava odabir promjenljivih i smanjenje dimenzionalnosti modela [48].

Kod *Lasso* algoritma, koeficijenti $\beta_j$ se procjenjuju korištenjem optimizacije koja minimizuje penalizovanu sumu apsolutnih vrijednosti koeficijenata regresije. Ova optimizacija može biti prikazana kao funkcija cijene sa (3.12), gdje je $\lambda$ parametar regularizacije, $\beta_j$ koeficijenti regresije, a $n$ broj posmatranja [45].

$$\text{Funkcija cijene: } \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \qquad (3.12)$$

*Lasso* regresija nudi prednost selekcije ključnih promjenljivih, što pojednostavljuje model i olakšava interpretaciju rezultata, što je korisno kada postoji veliki broj ulaznih promjenljivih. Ova metoda omogućava fokusiranje na najvažnije faktore, čime se poboljšava efikasnost modela. Međutim, *Lasso* regresija može ponekad eliminisati važne promjenljive koje su visoko korelirane s drugima, što može rezultovati gubitkom bitnih informacija i

smanjenom tačnošću predikcija. Takođe, u slučaju vrlo velikih skupova podataka, njena efikasnost može opasti, zahtijevajući dodatna prilagođavanja.

### 3.3.5. Slučajna šuma

Slučajna šuma (eng. *random forest*; RF) je ansambl algoritam koji koristi kombinaciju više stabala odluke za rješavanje zadataka klasifikacije i regresije. Ideja iza slučajne šume je da se agregiranjem rezultata iz više različitih stabala odluke smanji varijansa modela i poveća tačnost predikcija. Svako stablo u šumi gradi se korištenjem različitih podskupova podataka i podskupova promjenljivih, što doprinosi robustnosti modela prema prekomjernom prilagođavanju (eng. *overfitting*). Slučajna šuma je sposobna efikasno da radi sa velikim skupovima podataka sa mnogim prediktorima i često pruža mnogo bolje performanse u poređenju sa pojedinačnim stablima odluke [50-51].

Slučajna šuma koristi tehniku poznatu kao *bagging* za nasumično uzorkovanje podataka, čime se stvaraju različiti podskupovi iz originalnog trening skupa. Ovaj pristup povećava stabilnost i tačnost modela, omogućavajući bolju generalizaciju na nove podatke [52]. Svako stablo odluke unutar šume koristi nasumično odabrani podskup prediktora, što smanjuje korelaciju između stabala i smanjuje varijansu modela. Proces gradnje svakog stabla uključuje biranje optimalnih promjenljivih za podjele u čvorovima, čime se postiže veća raznolikost među stablima [47].

Nakon što su sva stabla izgrađena, njihove predikcije se kombinuju putem većinskog glasanja (za klasifikaciju) ili srednje vrijednosti (za regresiju) kako bi se dobila konačna predikcija. U slučaju regresije, konačna predikcija $\hat{y}$ se računa kao prosjek predikcija svih pojedinačnih stabala kao u (3.13), gdje je $T$ ukupan broj stabala u šumi, a $\widehat{y_t}$ predikcija pojedinačnog stabla $t$. U slučaju klasifikacije, konačna klasa se određuje putem većinskog glasanja, gdje svako stablo "glasa" za određenu klasu, a klasa sa najviše glasova postaje konačna predikcija [53].

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} \widehat{y_t} \qquad (3.13)$$

Slučajna šuma pruža robusne predikcije u radu s kompleksnim i raznovrsnim podacima, zahvaljujući svojoj sposobnosti da kombinuje informacije iz različitih stabala i procijeni važnost svake promjenljive. Ovaj algoritam je otporan na šum u podacima i dobro generalizuje, što ga čini pogodnim za mnoge primjene. Međutim, slučajna šuma može biti računski intenzivna, što produžava vrijeme obuke, naročito kod velikih skupova podataka. Njena kompleksnost može dodatno otežati interpretaciju rezultata, čineći analizu teže razumljivom [54].

### 3.3.6. *XGBoost*

*XGBoost* (eng. *extreme gradient boosting*) je napredna i optimizovana implementacija *gradient boosting* tehnike, specijalizovana za velike i složene skupove podataka. *Gradient boosting* je ansambl tehnika koja kombinuje predikcije nekoliko slabih modela, obično stabala odluke, kako bi formirala snažan model. *XGBoost* koristi tehniku *gradient boosting*-a za minimizaciju grešaka, čime postiže visok stepen prilagodljivosti i tačnosti. Zahvaljujući svojoj

brzini i efikasnosti, *XGBoost* je postao jedan od najpopularnijih alata u mašinskom učenju, posebno u kontekstu takmičenja i projekata koji zahtijevaju brzo i precizno modelovanje na velikim skupovima podataka [55].

*XGBoost* implementira niz optimizacijskih tehnika koje značajno poboljšavaju njegovu brzinu i performanse. Korištenjem paralelne obrade, algoritam omogućava istovremeno izračunavanje informacione dobiti (eng. *gain*) za više čvorova stabala, što rezultuje značajnim ubrzanjem procesa treniranja. Takođe, *XGBoost* efikasno upravlja memorijskim resursima zahvaljujući kompaktnim strukturama podataka i sofisticiranim algoritmima za obradu, čime se osigurava efikasnost čak i prilikom rada s velikim skupovima podataka. Uz ove tehnike, *XGBoost* se oslanja na specijalizovane algoritme za aproksimaciju i procjenu, koristeći pristupe zasnovane na histogramu, koji omogućavaju brzo pronalaženje optimalnih podjela u stablima. Ove napredne optimizacije čine *XGBoost* visoko efikasnim i prilagodljivim rješenjem za brojne zadatke u mašinskom učenju. *XGBoost* takođe podržava različite tipove podataka, uključujući numeričke i kategoričke podatke. Takođe nudi ugrađene metode za rukovanje nedostajućim vrijednostima i neuravnoteženim skupovima podataka, čime se dodatno povećava njegova primjenjivost u širokom spektru problema [56-57].

*XGBoost* koristi niz slabih modela (stabala odluke) koji se sekvencijalno treniraju, pri čemu svako novo stablo koriguje greške prethodnog. Osnovna ideja je da se minimizuje funkcija gubitka $L(\theta)$ dodavanjem novih stabala koja unapređuju prethodnu predikciju iz (3.14)[8], gdje je $\widehat{y_i}^{\,t}$ predikcija u $t$-om koraku, $\eta$ je stopa učenja, a $f_t(x_i)$ je predikcija novog stabla [58].

$$\widehat{y_i}^{\,t} = \widehat{y_i}^{\,(t-1)} + \eta f_t(x_i) \tag{3.14}$$

Optimizacija u *XGBoost*-u koristi *gradient boosting* tehniku koja minimizuje funkciju gubitka putem iterativne procedure. Funkcija gubitka $L(\theta)$ može biti srednjekvadratna greška za regresiju ili *log-loss* za klasifikaciju što je određeno sa (3.15), gdje je $\mathrm{l}(y_i, \widehat{y_i})$ funkcija gubitka koja mjeri razliku između stvarne vrijednosti $y_i$ i predikcije $\widehat{y_i}$.

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) \tag{3.15}$$

U *XGBoost*-u, dodavanje novog stabla $f_t(x_i)$ optimizuje se minimizacijom aproksimacije funkcije gubitka pomoću Tejlorovog razvoja do drugog reda. Proširenje je dato u (3.16), gdje su $g_i$ i $h_i$ prvi i drugi izvodi funkcije gubitka u odnosu na predikciju $\widehat{y_i}$, dok $\Omega(f_t)$ predstavlja regularizaciju koja pomaže u sprječavanju prekomjernog prilagođavanja [59].

$$L^{(t)}(\theta) \approx \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{3.16}$$

Konačna predikcija u *XGBoost*-u se dobija kombinovanjem predikcija svih stabala u modelu putem ponderisane sume prikazane u (3.17), gdje je $T$ ukupan broj stabala u modelu, a $\eta$ stopa učenja koja kontroliše doprinos svakog stabla konačnoj predikciji [47].

[8] Napomena: prilikom kreiranja promjenljive u formuli sa oznakom kape, došlo je do promjene indeksa "i" jer se tačka iznad slova gubi zbog formatiranja teksta. Ova promjena je tehničke prirode i ne utiče na tačnost ili značaj formule.

$$\hat{y}_i = \sum_{t=1}^{T} \eta f_t(x_i) \tag{3.17}$$

*XGBoost* je poznat po svojoj visokoj tačnosti i efikasnosti, posebno kada se radi s velikim i složenim skupovima podataka. Njegova sposobnost brze obrade i prilagođavanja različitim problemima čini ga idealnim za situacije gdje su potrebne brze i tačne predikcije. Ipak, *XGBoost* može biti izazovan za podešavanje zbog velikog broja hiperparametara koji zahtijevaju finu kontrolu, što može biti zahtjevno za korisnike bez iskustva. Takođe, složenost modela može otežati interpretaciju rezultata, što može ograničiti njegovu primjenu u situacijama gdje je transparentnost ključna.

### 3.3.7. *CatBoost*

*CatBoost* (eng. *categorical boosting*) je napredni algoritam *gradient boosting* tehnike, posebno razvijen za rad sa kategorizovanim podacima. Ovaj algoritam se ističe svojim specifičnim pristupom obradi kategorizovanih promjenljivih, što mu omogućava postizanje boljih performansi u poređenju sa drugim algoritmima bustinga. *CatBoost* smanjuje uticaj promjenljivih koje imaju visoku korelaciju i koristi optimizovane tehnike za smanjenje prekomjernog prilagođavanja. Njegova sposobnost da efikasno radi sa heterogenim podacima koji sadrže i numeričke i kategorizovane promjenljive čini ga izuzetno korisnim u različitim aplikacijama, kao što su finansije, marketing i industrijska analitika [60].

*CatBoost* se razlikuje od drugih algoritama bustinga prvenstveno po načinu na koji obrađuje kategorizovane podatke. Tradicionalni algoritmi često zahtijevaju unaprijed određeno kodiranje kategorija (npr. *one-hot encoding*), što može dovesti do gubitka informacija ili povećanja dimenzionalnosti. Nasuprot tome, *CatBoost* koristi statističke metode za pretvaranje kategorizovanih promjenljivih u numeričke vrijednosti na način koji zadržava korisne informacije. Ovaj proces obuhvata računanje ciljne vrijednosti unutar svake kategorije, dok se spriječava "curenje podataka" (eng. *data leakage*) kroz posebne tehnike kao što su *mean encoding* sa dodatkom šuma [61].

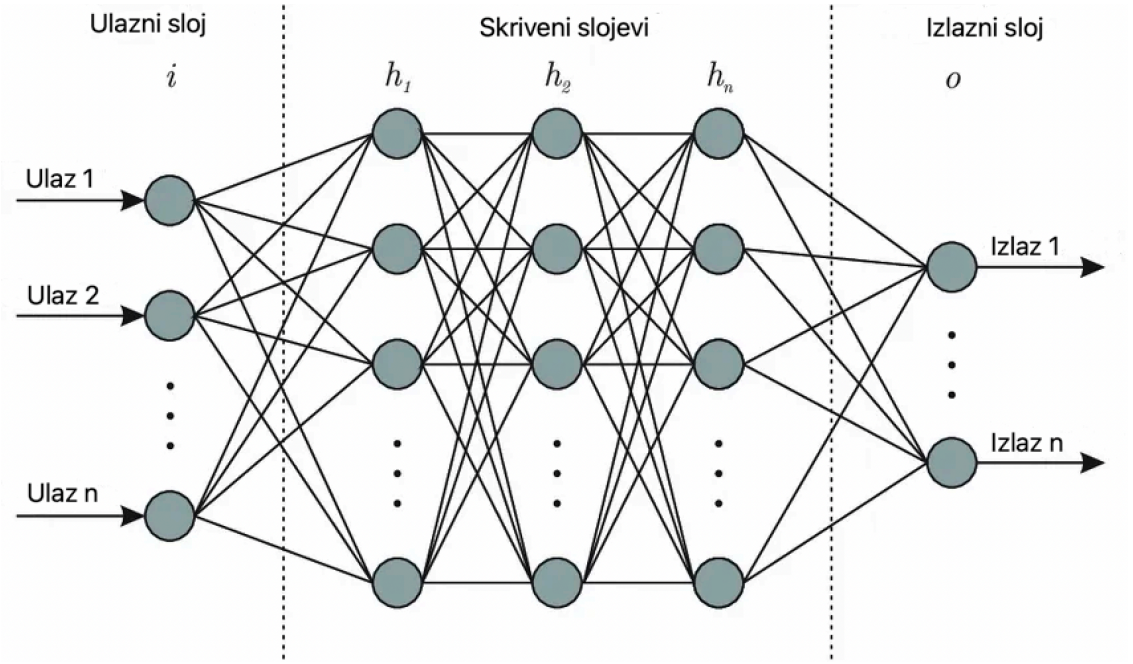
*CatBoost* kao i *XGBoost* ima podršku za paralelno procesiranje i podržava efikasnu distribuciju zadataka na računarskim klasterima čime se povećava efikasnost treniranja. *CatBoost* je dizajniran tako da bude robusniji prema prekomjernom prilagođavanju, zahvaljujući ugrađenim metodama za regulaciju kompleksnosti modela, poput L2 regularizacije i kontrole dubine stabala. Njegova otpornost na neuravnotežene skupove podataka i sposobnost rada sa kompleksnim heterogenim podacima čine ga izuzetno prilagodljivim za različite primjene [62-64].

*CatBoost* algoritam ima strukturu sličnu *XGBoost*-u, uključujući prikupljanje i pripremu podataka, sekvencijalno treniranje stabala odluke, te minimizaciju funkcije gubitka korištenjem *gradient boosting* tehnike. Međutim, ključna razlika je u obradi kategorizovanih promjenljivih, gdje *CatBoost* koristi posebne statističke metode za kodiranje koje minimizuju gubitak informacija. Takođe, *CatBoost* optimizuje funkciju gubitka uz dodatne tehnike prilagođene specifično za rad s kategorizovanim podacima, što ga čini efikasnijim u situacijama gdje se takvi podaci često pojavljuju.

*CatBoost* se ističe svojom sposobnošću efikasnog rukovanja kategoričkim podacima, što smanjuje proces pretprocesiranja i omogućava stabilne predikcije čak i u složenim uslovima. Ovaj algoritam je posebno koristan kada je potrebno raditi s velikim brojem različitih tipova podataka, jer njegovi optimizacijski procesi omogućavaju visoku tačnost. Međutim, *CatBoost* može biti resursno intenzivan, zahtijevajući više vremena i računske snage za obuku, što može predstavljati izazov u okruženjima sa ograničenim resursima. Njegova složenost takođe može otežati interpretaciju rezultata, što može biti problem kada je potrebna jasna analiza uticaja pojedinačnih promjenljivih.

## 3.4. Algoritmi dubokog učenja

Neuronske mreže su ključna komponenta dubokog učenja i koriste se za modelovanje složenih odnosa između ulaznih podataka i izlaznih predikcija. One su inspirisane strukturom i funkcijom ljudskog mozga i sastoje se od neurona organizovanih u slojeve. Svaki neuron prima ulaze, primjenjuje težine i aktivacionu funkciju, te generiše izlaz koji se prenosi dalje kroz mrežu. Opšta struktura neuronskih mreža je prikazana na Slika 3.2.



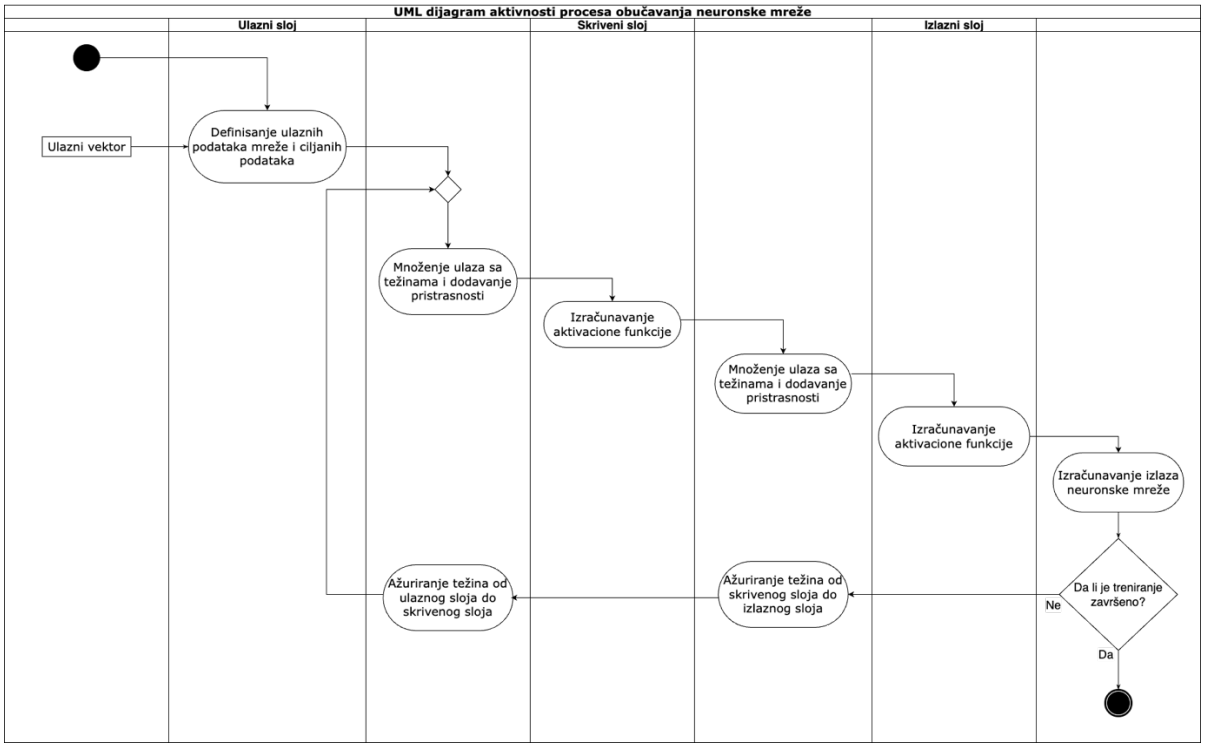

*Slika 3.2. Opšta struktura neuronskih mreža*

Ulazni sloj prima podatke i svaki njegov neuron predstavlja jednu ulaznu promjenljivu, poput piksela slike, senzorskih podataka ili drugih obilježja [65]. Podaci zatim prolaze kroz jedan ili više skrivenih slojeva, gdje se obrađuju i transformišu pomoću aktivacionih funkcija kao što su ReLU, *sigmoid* ili *tanh*, čime se uvodi nelinearnost i omogućava mreži da uči složene odnose među podacima. Svaki neuron u skrivenim slojevima prima informacije od svih neurona iz prethodnog sloja, vrši proračune koristeći pridružene težine i aktivacionu funkciju, i prosljeđuje obrađene informacije dalje. Na kraju, izlazni sloj generiše konačne predikcije na osnovu obrađenih podataka iz skrivenih slojeva. Broj neurona u izlaznom sloju varira u zavisnosti od zadatka koji se rješava; za klasifikaciju može imati jedan ili više neurona, dok za regresione zadatke može imati jedan neuron koji daje kontinuiranu vrijednost. Svi slojevi u mreži međusobno su povezani, i svaka veza ima pridruženu težinu koja se prilagođava tokom

procesa učenja, kako bi mreža mogla precizno predviđati ishode na osnovu ulaznih podataka. Ova fleksibilnost i sposobnost prilagođavanja čini neuronske mreže moćnim alatom za različite aplikacije u mašinskom učenju. Da bi se postigla tačna predviđanja, mreža mora optimizovati svoje težine, a to se postiže procesom poznatim kao gradijentni spust.

Gradijentni spust (eng. *gradient descent*) je algoritam optimizacije koji mreži omogućava da minimizuje funkciju greške prilagođavanjem težina veza između neurona. Funkcija greške mjeri razliku između stvarnih izlaza i onih koje mreža predviđa, i cilj je minimizovati tu razliku kako bi model bio što tačniji. Tokom procesa učenja, gradijentni spust izračunava gradijent, odnosno prvi izvod funkcije greške u odnosu na težine, i koristi te informacije da koriguje težine u smjeru u kojem greška opada. Ovaj iterativni proces se ponavlja sve dok greška ne bude svedena na minimum, čime mreža postaje sve bolja u predviđanju željenih izlaza na osnovu ulaznih podataka. Na ovaj način, gradijentni spust omogućava mreži da se efikasno prilagođava i uči iz podataka, usavršavajući svoje performanse tokom vremena.

Proces treniranja neuronskih mreža je prikazan na UML dijagramu aktivnosti prikazanom na Slici 3.3.




*Slika 3.3. UML dijagram aktivnosti procesa obučavanja neuronske mreže*

Proces započinje inicijalizacijom težina i pristranosti, koristeći nasumične vrijednosti ili tehnike kao što su *Xavier* ili *He* inicijalizacija, kako bi se osiguralo efikasno učenje [66]. Nakon inicijalizacije, mreža ulazi u fazu prolaska unaprijed, gdje se definisani ulazni podaci i ciljani izlazi koriste za računanje izlaznih vrijednosti. Tokom ovog koraka, podaci prolaze kroz mrežu, počevši od ulaznog sloja, gdje se ulazi množe sa težinama i dodaje se pristrasnost. Ovi proračuni se nastavljaju kroz skriveni sloj, gdje se primjenjuju aktivacione funkcije koje uvode nelinearnost i omogućavaju mreži da uči složene odnose među podacima.

Nakon prolaska unaprijed, mreža izračunava funkciju gubitka, mjeri razliku između stvarnih i prediktovanih vrijednosti, koristeći funkcije gubitka poput *cross-entropy* za klasifikaciju ili srednje kvadratne greške za regresiju. Ovaj korak omogućava mreži da procijeni tačnost svojih predviđanja. Slijedi propagacija unazad, tehnika koja računa gradijente funkcije gubitka u odnosu na težine i pristrasnosti, koristeći lančano pravilo. Propagacija unazad omogućava mreži da koriguje greške nazad kroz slojeve, od izlaznog do skrivenog i ulaznog sloja, ažurirajući težine tako da smanjuje grešku.

Tokom ovog procesa, težine se ažuriraju korištenjem optimizacijskih algoritama kao što su gradijentni spust ili *adam*. Ovi algoritmi koriste izračunate gradijente da bi prilagodili težine, čime se mreža poboljšava u učenju i tačnosti predviđanja. Nakon ažuriranja težina od ulaznog do skrivenog sloja, proces se ponavlja sve dok mreža ne postigne željenu tačnost. Ako je treniranje završeno, mreža zaustavlja proces prilagođavanja; u suprotnom, vraća se na početak i ponavlja ciklus učenja. Ovaj iterativni proces omogućava mreži da kontinuirano poboljšava svoje performanse, prilagođavajući svoje parametre kroz svakodnevne ponovljene korake, sve dok se ne postignu optimalni rezultati.

Postoji nekoliko tipova neuronskih mreža, od kojih svaka ima specifične karakteristike i primjene. U ovoj sekciji su detaljno opisani ključni tipovi neuronskih mreža korištenih za predikciju zdravlja litijum-jonskih baterija, a to su potpuno povezane neuronska mreža, LSTM kao predstavnik rekurentne neuronske mreže i konvoluciona neuronska mreža.

### 3.4.1. Potpuno povezana neuronska mreža

Potpuno povezana neuronska mreža (eng. *fully connected neural network*; FCNN), poznata i kao višeslojni perceptron (eng. *multilayer perceptron*; MLP), je osnovni tip neuronske mreže koji se sastoji od ulaznog sloja, nekoliko skrivenih slojeva i izlaznog sloja. U ovoj mreži, svaki neuron u jednom sloju povezan je sa svim neuronima u sljedećem sloju, što omogućava mreži da modeluje složene nelinearne odnose između ulaznih i izlaznih podataka. Potpuno povezane neuronske mreže se koriste za širok spektar problema, uključujući klasifikaciju i regresiju, i treniraju se korištenjem algoritma propagacije unazad (eng. *backpropagation*), koji optimizuje težine neurona minimizovanjem greške predikcije [67-68].

Na Slici 3.2 je prikazana opšta struktura potpuno povezane neuronske mreže. Pored opštih informacija o ovoj neuronskoj mreži važno je pokazati kako se izlaz iz svakog skrivenog sloja može matematički opisati sa (3.18), gdje je $h^{(l)}$ izlaz iz sloja $l$, $W^{(l)}$ matrica težina za sloj $l$, $b^{(l)}$ ofset, a $\phi$ aktivaciona funkcija.

$$h^{(l)} = \phi\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \tag{3.18}$$

Tip aktivacione funkcije u posljednjem sloju zavisi od prirode problema. Na primjer, *softmax* funkcija se često koristi za klasifikaciju, dok se linearna aktivaciona funkcija koristi za regresiju. Za klasifikaciju, izlaz iz izlaznog sloja može biti opisan u (3.19), gdje $\hat{y}$ predstavlja vektor estimiranih vjerovatnoća da uzorak pripada svakoj od klasa.

$$\hat{y} = softmax\left(W^{(L)}h^{(L-1)} + b^{(L)}\right) \tag{3.19}$$

Proces treniranja potpuno povezane neuronske mreže ilustrovan je dijagramom aktivnosti na Slici 3.3, koji jasno prikazuje ključne korake učenja. Nakon prolaska unaprijed kroz mrežu i generisanja predikcija, mreža se suočava sa zadatkom procjene svojih performansi.

Mjerenje razlike između stvarnih i predikovanih vrijednosti vrši se pomoću funkcije gubitka, koja igra važnu ulogu u evaluaciji mreže. Za zadatke klasifikacije, često se koristi *cross-entropy* funkcija gubitka koja efektivno mjeri neslaganje između stvarnih klasa i predikcija. S druge strane, za regresione zadatke, srednja kvadratna greška (MSE) kao što je prikazano sa (3.20), gdje je $L$ ukupna greška, $y_i$ stvarna vrijednost, a $\widehat{y_i}$ predikcija.

$$L = \frac{1}{n}\sum_{i=1}^{n} MSE(y_i, \widehat{y_i}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \qquad (3.20)$$

Težine i pristranosti se ažuriraju korištenjem optimizacijskih algoritama kao što su gradijentni spust ili *adam*. Novi parametri se izračunavaju kao što je prikazano sa (3.21), koristeći gradijente iz propagacije unazad, gdje je $\eta$ stopa učenja, a $\frac{\partial L}{\partial W^{(l)}}$ gradijent funkcije gubitka u odnosu na težine [69].

$$W^{(l)} = W^{(l)} - \eta \frac{\partial L}{\partial W^{(l)}} \qquad (3.21)$$

Potpuno povezane neuronske mreže su korisne za predikciju zdravlja litijum-jonskih baterija zbog svoje sposobnosti da uče složene nelinearne odnose između različitih mjerenja, poput napona, struje i temperature, i stanja zdravlja baterija. Njihova fleksibilnost omogućava modelima da se prilagode specifičnostima podataka o baterijama, pružajući precizne i pouzdane predikcije. Međutim, s obzirom na to da ove mreže povezuju sve neurone iz jednog sloja sa svim neuronima u sljedećem, može doći do prekomjernog prilagođavanja, posebno kada se koriste sa velikim brojem ulaznih promjenljivih i ograničenim skupom podataka. Ovo može rezultovati modele koji dobro funkcionišu na trening podacima, ali gube tačnost pri predikciji novih, neviđenih podataka. S druge strane, zbog svoje jednostavnosti, potpuno povezane neuronske mreže se lako implementiraju i mogu efikasno obraditi različite tipove podataka, što ih čini pogodnim za analizu i praćenje zdravlja baterija u realnom vremenu. Uprkos potencijalnim izazovima sa skalabilnošću, njihova robusnost i sposobnost generalizacije čine ih korisnim alatom u održavanju i upravljanju baterijama [70].

### 3.4.2. Rekurentna neuronska mreža

Rekurentna neuronska mreža (eng. *recurrent neural network*; RNN) je specijalni tip neuronske mreže dizajniran za obradu sekvencijalnih podataka. RNN-ovi se razlikuju od potpuno povezanih neuronskih mreža po tome što imaju ciklične veze, koje omogućavaju prenošenje informacija kroz različite korake u sekvenci. Ove mreže su posebno pogodne za zadatke koji uključuju sekvencijalne ili vremenske podatke, kao što su analiza vremenskih serija, prepoznavanje govora i obrada prirodnog jezika. Glavna prednost RNN-ova je njihova sposobnost da hvataju zavisnosti u vremenskim ili sekvencijalnim podacima, što omogućava bolje modeliranje složenih obrazaca u podacima [71-74].

Prvi sloj mreže koji prima sekvencijalne ulazne podatke. Svaki neuron u ovom sloju predstavlja jednu ulaznu promjenljivu u svakoj tački vremena (eng. *timestep*). Na primjer, ako

mreža obrađuje vremensku seriju sa $n$ ulaznih promjenljivih, ulazni sloj će imati $n$ neurona za svaki trenutak vremena $t$.

Jedan ili više slojeva sa cikličnim vezama, gdje svaka jedinica u sloju prima ulaz iz prethodnog sloja i iz prethodnog trenutka vremena. Ovi slojevi omogućavaju mreži da pamti informacije iz prethodnih vremenskih koraka, čime se hvataju zavisnosti između različitih tačaka u sekvenci. Aktivnost u skrivenom sloju može se opisati sa (3.22), gdje je $h^{(t)}$ skriveno stanje u trenutku $t$, $W_{xh}$ matrica težina za ulazne podatke, $W_{hh}$ matrica težina za povratne veze (eng. *recurrent connections*), $x^{(t)}$ ulaz u trenutku $t$, a $b_h$ vektor pristranosti.

$$h^{(t)} = \phi\left(W_{xh} \cdot x^{(t)} + W_{hh} \cdot h^{(t-1)} + b_h\right) \qquad (3.22)$$

Posljednji sloj mreže koji generiše izlazne predikcije za svaki trenutak vremena $t$ . Izlaz iz ovog sloja može zavisiti od trenutnog stanja $h^{(t)}$ i može se opisati sa (3.23), gdje je $\hat{y}^{(t)}$ predikcija u trenutku $t$, $W_{hy}$ matrica težina za izlazni sloj, a $b_y$ vektor pristranosti.

$$\hat{y}^{(t)} = \phi\left(W_{hy}h^{(t)} + b_y\right) \qquad (3.23)$$

Rekurentne neuronske mreže zahtijevaju poseban pristup treniranju zbog svoje sposobnosti da obrađuju sekvencijalne podatke i pamte informacije kroz vrijeme. Nakon inicijalizacije težina i pristranosti, mreža prolazi kroz fazu prolaska unaprijed, gdje se koriste trenutne težine i pristranosti za izračunavanje izlaznih vrijednosti za svaki ulaz u sekvenci. Ovaj proces omogućava mreži da obrađuje podatke kroz vremensku dimenziju, uzimajući u obzir zavisnosti iz prethodnih vremenskih koraka. Kako bi mreža mogla prilagoditi svoje težine i poboljšati performanse, izračunava se funkcija gubitka koja mjeri razliku između stvarnih i predikovanih vrijednosti za svaku tačku u sekvenci, koristeći funkcije poput *cross-entropy* za klasifikaciju ili srednje kvadratne greške za regresiju.

Propagacija unazad kroz vrijeme (eng. *backpropagation through time*; BPTT) omogućava mreži da računa gradijente funkcije gubitka u odnosu na težine i pristranosti kroz cijelu sekvencu, koristeći lančano pravilo za prilagođavanje. Ovaj proces je proširenje standardne unazadne propagacije, prilagođeno za RNN-ove, čime se osigurava učenje iz grešaka tokom cijele sekvence. Ovi koraci omogućavaju rekurentnim mrežama da efikasno uče iz sekvencijalnih podataka, poboljšavajući svoje sposobnosti u modeliranju vremenskih obrazaca i odnosa.

RNN-ovi su izuzetno korisni za predikciju zdravlja litijum-jonskih baterija jer mogu efikasno modelovati sekvencijalne podatke, kao što su ciklusi punjenja i pražnjenja. Njihova sposobnost da hvataju vremenske zavisnosti omogućava preciznije predikcije stanja zdravlja baterija na osnovu istorijskih podataka. Ovo je posebno važno za baterije, gdje promjene u stanju zdravlja često zavise od prethodnih ciklusa upotrebe i uslova rada, čineći RNN-ove pogodnim alatom za ovakve zadatke [75].

### 3.4.3. Konvoluciona neuronska mreža

Konvoluciona neuronska mreža je specijalizovana arhitektura neuronskih mreža dizajnirana za obradu podataka sa prostornom strukturom, poput slika, video zapisa i drugih podataka sa prostornim ili vremenskim strukturama. CNN-ovi su posebno korisni za zadatke

koji uključuju prepoznavanje obrazaca u podacima, kao što su klasifikacija slika, segmentacija, detekcija objekata i prepoznavanje lica. Ključna karakteristika CNN-ova je upotreba konvolucionih slojeva, koji primjenjuju filtere za detekciju lokalnih obrazaca u ulaznim podacima [76].

Prvi sloj mreže koji prima ulazne podatke, najčešće u obliku slike. Ako se radi sa slikama, ulazni sloj obično ima dimenzije $H \times W \times C$ , gdje su $H$ visina, $W$ širina slike, a $C$ broj kanala (npr. tri za RGB slike).

Konvolucioni slojevi primjenjuju filtere za detekciju lokalnih obrazaca u ulaznim podacima. Svaki filter (jezgro) klizi preko ulaznih podataka, izračunavajući unutrašnji proizvod između filtera i lokalnog regiona ulaznih podataka. Rezultat ove operacije je mapa karakteristika (eng. *feature map*), koja naglašava prisustvo određenih karakteristika u različitim dijelovima slike. Operacija konvolucije može se matematički izraziti sa (3.24), gdje je $z_{i,j,k}$ vrijednost u mapi karakteristika na poziciji $(i, j)$ za $k$-ti filter, $x$ je ulazna slika, $w$ je filter, a $b_k$ je pristranost za $k$-ti filter.

$$z_{i,j,k} = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{c=1}^{C} x_{i+m-1,j+n-1,c} \cdot w_{m,n,c,k} + b_k \qquad (3.24)$$

Slojevi agregacije (eng. *pooling layers*) smanjuju dimenzionalnost podataka putem operacija kao što su maksimalno (eng. *max pooling*) i srednje (eng. *average pooling*) spuštanje. Ovi slojevi pomažu u smanjenju broja parametara i povećavaju robusnost mreže na varijacije u ulaznim podacima. Na primjer, u *max pooling* operaciji, za svaku oblast u mapi karakteristika, uzima se maksimalna vrijednost, čime se smanjuje dimenzionalnost kao što je dato u (3.25), gdje $z_{i,j,k}$ predstavlja izlaz iz *pooling* sloja za $k$-tu mapu karakteristika.

$$z_{i,j,k} = \max_{m,n} \{ z_{i+m-1,j+n-1,k} \} \qquad (3.25)$$

Potpuno povezani slojevi (eng. *fully connected layers*) povezuju sve neurone iz prethodnog sloja sa svim neuronima u sljedećem sloju. Nakon konvolucionih i *pooling* slojeva, podaci se često izravnavaju (eng. *flattening*) u jedan vektor, koji se zatim prosljeđuje kroz jedan ili više potpuno povezanih slojeva. Ovi slojevi obično nalaze na kraju mreže i koriste se za klasifikaciju ili regresiju. Formula za potpuno povezani sloj je data sa (3.26), gdje je $y$ izlazni vektor, $W$ matrica težina, $x$ ulazni vektor, $b$ pristranost, a $\phi$ aktivaciona funkcija (npr. ReLU, sigmoid).

$$y = \phi(Wx + b) \qquad (3.26)$$

Posljednji sloj mreže koji generiše izlazne predikcije. Za klasifikacione zadatke, izlazni sloj često koristi *softmax* funkciju za pretvaranje izlaza u vjerovatnoće za svaku klasu je data sa (3.27), gdje je $\widehat{y_i}$ vjerovatnoća za klasu i, $z_i$ aktivacija iz prethodnog sloja, a $K$ ukupan broj klasa [76].

$$\widehat{y_i} = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)} \qquad (3.27)$$

Proces treniranja konvolucionih neuronskih mreža je vrlo sličan procesu treniranja potpuno povezanih mreža. Na Slici 3.3 je prikazan detaljan način na koji se ovaj proces odvija.

CNN-ovi mogu ukazivati na degradaciju materijala unutar litijum-jonskih baterija kroz prepoznavanje i analizu obrazaca u podacima koji odražavaju promjene u elektrohemijskim svojstvima baterije. Na primjer, kako baterija stari, određeni parametri poput kapaciteta i unutrašnje otpornosti mogu se mijenjati. CNN-ovi prepoznaju ove promjene, koje mogu biti znakovi procesa degradacije, kao što su rast dendrita ili propadanje elektrolita. Na taj način, CNN-ovi omogućavaju preciznije predikcije stanja zdravlja baterije, ukazujući na unutrašnje promjene koje bi inače bile teško uočljive [77].

# 4. Korišteni skupovi podataka

## 4.1. NASA skup podataka

Podaci korišteni u ovom master radu potiču iz opsežnih eksperimenata sa litijum-jonskim baterijama sprovedenih u *NASA Prognostics Center of Excellence* [78]. Eksperimenti su realizovani u različitim unutrašnjim i vanjskim uslovima okoline. Prikupljeni podaci o baterijama su organizovani u pojedinačne datoteke, pri čemu se svaka datoteka odnosi na jednu bateriju i sačuvana je u MATLAB formatu. Ovaj format omogućava efikasno skladištenje i obradu velikih količina podataka, što je ključno za duboke analize koje su sprovedene.

Skup podataka obuhvata informacije o 24 litijum-jonske baterije, ukupno 2881 ciklusa punjenja, pražnjenja i mjerenja impedanse, kako bi se procijenile njihove performanse i degradacije pod različitim uslovima. Proces punjenja je obično izvršen u režimu konstantne jačine struje od 1.5 A do dostizanja napona od 4.2 V, nakon čega se prelazilo na režim konstantnog napona sve dok jačina struje punjenja nije pala na 20 mA. Ovaj način punjenja je odabran kako bi se maksimizovala efikasnost punjenja i minimizovala degradacija ćelija. Procesi pražnjenja su varirali, uključujući nivoe konstantne struje, profil opterećenja kvadratnim talasima sa različitim amplitudama i radnim ciklusima, te nekoliko fiksnih nivoa opterećenja strujom, sa zaustavljanjem na određenim naponskim pragovima. Mjerenja impedanse su izvršena pomoću elektrohemijske spektroskopije impedanse (EIS) u frekvencijskom opsegu od 0.1 Hz do 5 kHz za sve skupove baterija. Korištena je logaritamska raspodjela frekvencija, sa 50 do 100 mjernih tačaka po dekadi, što omogućava preciznu analizu različitih procesnih skala unutar baterije. Ova tehnika omogućava detaljnu analizu unutrašnjeg stanja baterija, uključujući procjenu degradacije elektroda i elektrolita.

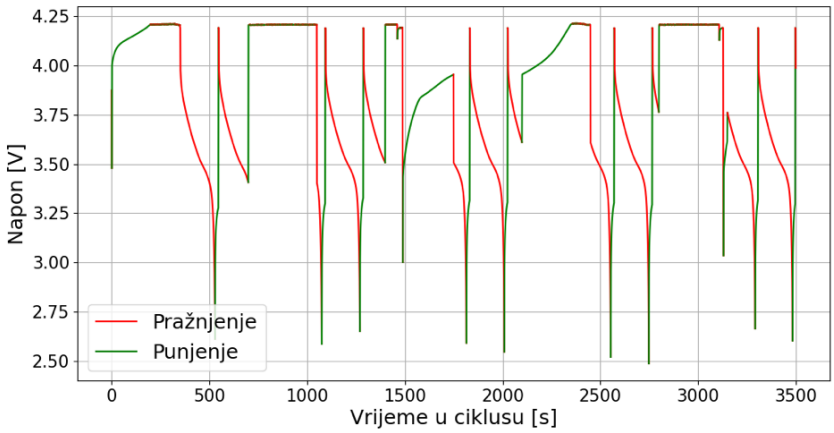
Eksperimenti su izvođeni pri različitim temperaturama okoline, u rasponu od sobne temperature do povišenih temperatura od 43 °C i niskih temperatura od 4 °C. Ove temperaturne varijacije su pažljivo odabrane kako bi simulirale stvarne uslove u kojima se baterije mogu koristiti, uključujući ekstremne klimatske uslove. Kriterijumi za kraj životnog vijeka ovih baterija bili su zasnovani na određenom smanjenju kapaciteta, bilo 20% ili 30%, sa nekoliko eksperimenata koji su bilježili izuzetno niske kapacitete ili su bili prekinuti zbog problema sa kontrolnim softverom. Ovo smanjenje kapaciteta je ključno za razumijevanje kako se baterije ponašaju tokom dugotrajnog cikličnog korištenja i pod različitim stresnim uslovima.

Struktura podataka bilježi detalje za svaki ciklus, uključujući tip operacije (punjenje, pražnjenje, impedansa), temperaturu okoline, vrijeme i mjerenja poput napona, struje, temperature, kapaciteta i raznih parametara vezanih za impedansu. Ovaj skup podataka pruža dragocjene uvide u ponašanje i degradaciju baterija pod različitim operativnim uslovima. Na Slici 4.1 je prikazan obrazac napona tokom ciklusa punjenja i pražnjenja litijum-jonske baterije, kako je zabilježeno u datoteci B0005.mat. Ovi grafici omogućavaju vizualizaciju i analizu ključnih karakteristika ciklusa, kao što su stabilnost napona i efikasnost punjenja i pražnjenja.

Grafikon na Slici 4.2 prikazuje trend gdje se stanje zdravlja baterije (SOH) smanjuje sa povećanjem vremena pražnjenja, odražavajući tipičan proces starenja baterija usljed

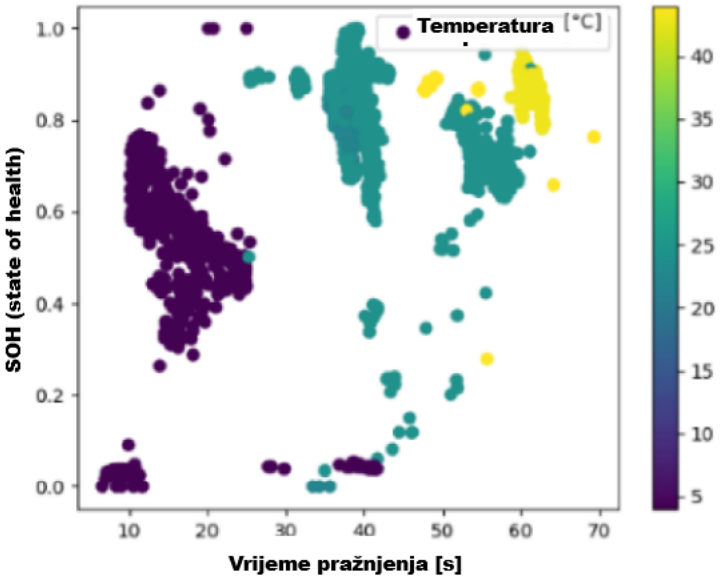
ponovljenih ciklusa punjenja i pražnjenja. Ovi trendovi su od suštinske važnosti za razvoj prediktivnih modela koji mogu pomoći u predviđanju preostalog vijeka trajanja baterija i optimizaciji njihovog korištenja u realnim aplikacijama.

Vrijeme pražnjenja i temperatura nisu jedini faktori koji utiču na SOH. Osim njih ostali faktori su izmjerena vrijednost napona, napon pražnjenja, izmjerena struja i struja pražnjenja takođe imaju ključnu ulogu. Kapacitet, koji je osnovni kriterijum za određivanje SOH, je stoga pod uticajem ovih električnih parametara. Gradijent boje na grafiku ukazuje da temperatura utiče na SOH, pri čemu su više temperature generalno povezane sa višim vrijednostima SOH. Međutim, čini se da odnos između temperature i SOH nije linearan, što sugeriše kompleksnu interakciju među različitim parametrima koji upravljaju zdravljem baterija.




*Slika 4.1. Ciklusi punjenja i pražnjenja baterije*

Ovaj pregled podataka i eksperimenata pruža ključne uvide u ponašanje i dugotrajnost litijum-jonskih baterija, kao i osnovu za razvoj naprednih modela za procjenu njihovih performansi i stanja zdravlja. Ove informacije su od suštinskog značaja za unapređenje tehnologije baterija, optimizaciju njihove upotrebe, i razvoj novih generacija energetskih sistema sa poboljšanom pouzdanosti. Detaljni uvidi omogućavaju inženjerima i istraživačima da razviju efikasnije strategije za održavanje i optimizaciju baterija, čime se povećava njihova dugovječnost i efikasnost u različitim aplikacijama [20].




*Slika 4.2. Zavisnost SOH metrike od temperature i vremena pražnjenja*

U poglavlju 5.1 opisan je eksperiment i prikazani su rezultati primjene regresije na NASA skup podataka. U poglavlju 5.2 opisan je eksperiment u kojem se koristi CNN za procjenu efekta vremenske informacije na predikciju unutar ciklusa.

### 4.1.1. Podjela i pretprocesiranje podataka

U okviru pripreme podataka za analizu, podaci za svaku bateriju nalaze se u odvojenim datotekama. Tokom ove faze, te datoteke se nasumično dijele u dva skupa: skup za obuku i skup za testiranje. Konkretno, 80% datoteka se koristi za obuku modela, dok se preostalih 20% koristi za testiranje. Ova podjela omogućava da podaci iz jedne baterije budu prisutni samo u jednom skupu, čime se izbjegava preklapanje i obezbjeđuje se realistična procjena modela na potpuno novim podacima.

U procesu pretprocesiranja, sadržaj svake datoteke se učitava i obrađuje kako bi se osigurala kvaliteta i integritet podataka za dalju analizu. Podaci se preuređuju i transformišu koristeći biblioteku *Pandas* u *Python* programskom jeziku. Ovo uključuje kreiranje karakteristika na osnovu statistika struje i napona tokom svakog ciklusa punjenja i pražnjenja, kao što su standardna devijacija, srednja vrijednost, minimalne i maksimalne vrijednosti. Ovi statistički parametri omogućavaju uniformnost i prilagođenost podataka za analizu.

Podaci koji se odnose na impedansu i punjenje baterija nisu uključeni u ovu analizu. Iako bi njihovo uključivanje moglo obezbijediti širu perspektivu i povećati opštost studije, ovaj pristup nije neuobičajen u literaturi. Mnoge studije se fokusiraju isključivo na cikluse pražnjenja, jer oni pružaju ključne informacije o trenutnom kapacitetu baterija, koji je ključni pokazatelj njihovog stanja. Ključna karakteristika skupa podataka je kapacitet baterije, koji je neophodan za procjenu zdravlja baterije. Tokom čišćenja podataka, kapacitet se konvertuje u stanje zdravlja baterije, što postaje ciljna promjenljiva za analizu i modeliranje. Maksimalni kapacitet korišten u ovoj studiji je 2 Ah. Za računanje SOH-a korištena je (2.1).

Različite studije su koristile slične metode za procjenu SOH baterija, ističući jednostavnost i efikasnost ovih pristupa. Iako u nekim slučajevima mogu biti korištene složenije metode, ova studija se oslanja na jednostavan i direktan pristup za procjenu stanja zdravlja baterije.

## 4.2. *Toyota* skup podataka

Ovaj skup podataka korišten je u studiji [28]. Koriste se 124 litijum-jonske baterije, koje su testirane dok nisu postale neupotrebljive pod uslovima brzog punjenja. Ove baterije proizvedene su od strane kompanije *A123 Systems* i označene su serijskim brojem *APR18650M1A*.

Baterije su testirane na uređaju *Arbin LBT* potenciostat. Potenciostat je uređaj koji se koristi u elektrohemiji za kontrolu i mjerenje napona i struje u elektrohemijskim ćelijama. Pomaže naučnicima da proučavaju reakcije koje se dešavaju na površini elektroda kada se na njih primijeni određeni napon. Omogućava precizno kontrolisanje uslova u eksperimentima gdje se proučavaju hemijske reakcije na elektrodama. Potenciostat je postavljen u komoru sa prisilnim protokom vazduha kako bi se temperatura održavala na 30°C. Svaka baterija ima nominalni kapacitet od 1,1 Ah (što znači da može isporučiti struje jačine od 1,1 A u toku 1 časa prije nego što se isprazni) i nominalni napon od 3,3 V.

U nastavku se koristi *C-rate*, pa ga je potrebno prethodno opisati. *C-rate* je mjera koja pokazuje koliko brzo se baterija prazni u odnosu na njen maksimalni kapacitet. Ova stopa normalizuje pražnjenje, što omogućava lakše poređenje različitih baterija, bez obzira na njihov kapacitet. Na primjer, *C-rate* od 1C znači da će se baterija isprazniti potpuno za jedan sat. Za bateriju koja ima kapacitet od 100 ampera-sati, to bi značilo struju pražnjenja od 100 ampera. Ako je *C-rate* 5C, to bi odgovaralo struji pražnjenja od 500 ampera, dok bi C/2 stopa značila struju pražnjenja od 50 ampera. Korištenje *C-rate* omogućava preciznije upravljanje brzinom pražnjenja baterije u različitim situacijama [79].

Cilj prethodno opisanog testiranja baterija je bio da se pronađu načini za optimizaciju brzog punjenja litijum-jonskih baterija. Brzo punjenje je proces koji može brže dovesti do degradacije baterija. U ovom skupu podataka, sve baterije su punjene koristeći jednu od dvije metode brzog punjenja. Ove metode su označene kao “C1(Q1)-C2”, gdje su C1 i C2 različiti nivoi konstantne struje tokom punjenja, a Q1 je stanje napunjenosti baterije izraženo u procentima (%). Drugi korak punjenja završava se kada baterija dostigne 80% napunjenosti, nakon čega se baterije pune pri 1C (što znači da se pune za jedan sat) konstantnom strujom i naponom (CC-CV). Gornji i donji granični naponi za punjenje su 3,6 V i 2,0 V, što je u skladu sa specifikacijama proizvođača.

Skup podataka je podijeljen u tri grupe, svaka sa oko 48 baterija. Grupe su definisane prema datumu kada su testovi započeti. Svaka grupa ima neke specifične nepravilnosti, koje su detaljno opisane u dokumentaciji za svaku grupu.

Mjerenja temperature su izvedena pričvršćivanjem termopara tipa T na baterije pomoću termalnog epoksida (specijalnog ljepila otpornog na visoke temperature) i Kapton trake (specijalne trake otporne na toplotu). Termopar tipa T je senzor koji se koristi za mjerenje

temperature. Napravljen je od dvije različite metalne žice, bakra i konstantana (legura bakra i nikla), koje su spojene na jednom kraju. Kada se ovaj spoj zagrije ili ohladi, stvara se električni napon (termoelektrični napon) koji se može mjeriti i koji je proporcionalan temperaturi. Termopar tipa T je poznat po svojoj preciznosti i stabilnosti, posebno pri niskim temperaturama, i obično se koristi u aplikacijama gdje su potrebna tačna mjerenja u rasponu od -200°C do 350°C. Ipak, mjerenja temperature nisu uvek potpuno pouzdana, jer kontakt između termopara i baterije može varirati, a termopar ponekad gubi kontakt tokom ciklusa punjenja i pražnjenja.

Unutrašnja otpornost baterija mjerena je tokom punjenja kada su baterije dostigle 80% napunjenosti, i to prosjekom 10 impulsa struje od ±3,6C sa širinom impulsa od 30 ms ili 33 ms, u zavisnosti od datuma mjerenja.

| | summary |
|---|---|
| cycle_index | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, … |
| discharge_capacity | [1.9529131999999998, 1.0661633, 1.0704726, 1.0717603999… |
| charge_capacity | [1.4413344, 1.0662235, 1.0702372, 1.0714923, 1.07223250… |
| discharge_energy | [6.182999099999999, 3.2448785, 3.2616343, 3.2637928, 3.… |
| charge_energy | [4.755055400000001, 3.7143865, 3.7208915, 3.72342799999… |
| dc_internal_resistance | [0.029383093118667603, 0.017350584268569946, 0.01715334… |
| temperature_maximum | [33.98908615112305, 36.437530517578125, 36.909210205078… |
| temperature_average | [30.05450439453125, 32.368080139160156, 32.767017364501… |
| temperature_minimum | [27.866647720336914, 30.063962936401367, 30.41473770141… |
| date_time_iso | [2017-07-01T04:05:20+00:00, 2017-07-02T01:33:52+00:00, … |
| energy_efficiency | [1.3003001184802174, 0.8735974298851238, 0.876573342705… |
| charge_throughput | [1.4413343667984009, 2.5075578689575195, 3.577795028686… |
| energy_throughput | [4.7550554275512695, 8.469442367553711, 12.190333366394… |
| charge_duration | [33024.0, 640.0, 640.0, 640.0, 640.0, 640.0, 512.0, 512… |
| time_temperature_integrated | [38666.01721191406, 1933.5312906901042, 1953.8413492838… |
| paused | [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, … |
| voltage | NaN |
| temperature | NaN |
| internal_resistance | NaN |
| current | NaN |
| step_type | NaN |

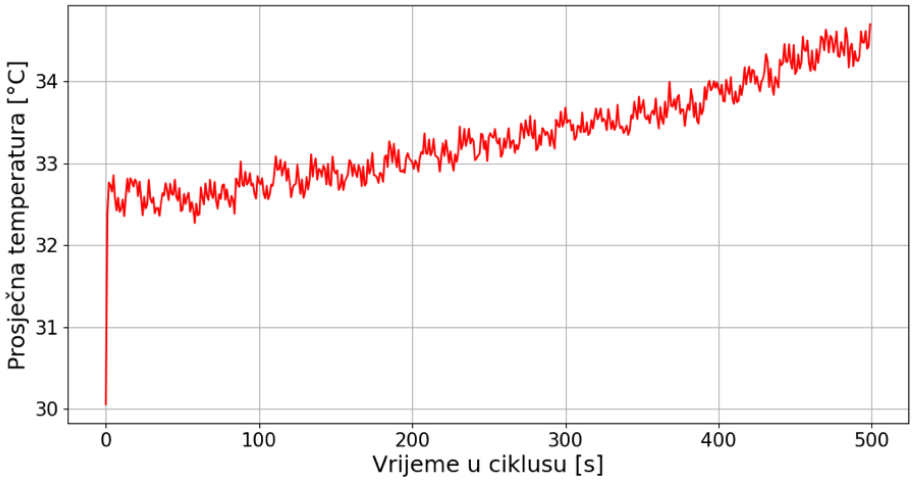
*Slika 4.3. Pregled kolona skupa podataka*

Podaci su dostupni u tri formata: MATLAB struktura (poseban tip datoteke koji koristi program MATLAB), CSV fajlovi (obični tekstualni fajlovi sa podacima odvojenim zarezima) i BEEP struktuirani format (JSON fajlovi). MATLAB struktura omogućava lak pristup podacima za svaki ciklus punjenja, dok CSV fajlovi ponekad sadrže greške u vremenu testa i vremenu koraka, koje su ispravljene u MATLAB strukturi.

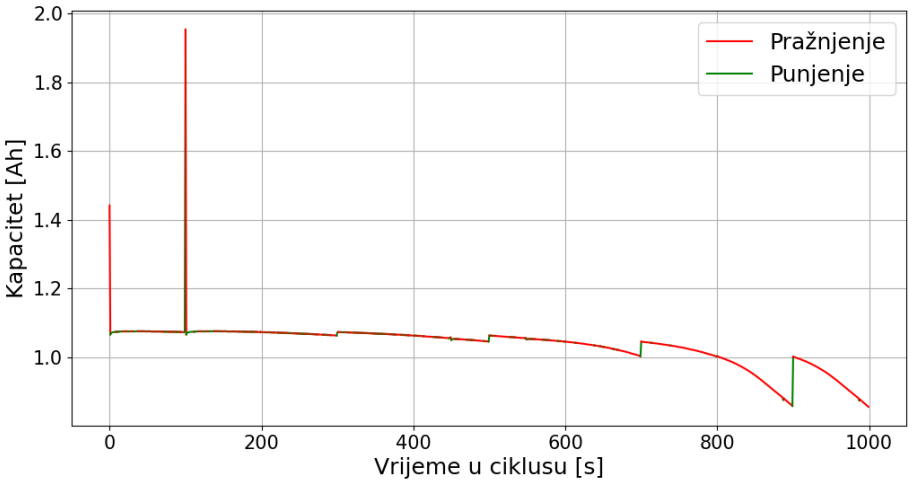
*BEEP* struktuirani format JSON fajlova predstavlja pogodan format za programsko učitavanje podataka. Veličina komprimovane arhive iznosi 27,95 GB. U tom skupu podataka nalazi se mnogo metapodataka koji nisu relevantni za primjenu u oblasti mašinskog učenja. Ključni elementi su podaci iz reda *summary*. Ovaj red sadrži višestruke nizove podataka koji zavise od kolone. Postoji značajan broj kolona od važnosti. Kolone su prikazane na Slici 4.3; međutim, važno je napomenuti da je prikazana transponovana matrica podataka radi boljeg vizuelnog prikaza, tako da redovi sa slike zapravo predstavljaju kolone. Svaki element u toj

matrici predstavlja odgovarajući niz podataka. Elementi nizova sa istim indeksom predstavljaju podatke za određeni ciklus baterije.

Kolone prikazane na Slici 4.3 predstavljaju sljedeće podatke: indeks ciklusa, kapacitet pražnjenja, kapacitet punjenja, energija pražnjenja, unutrašnji otpor, maksimalna, prosječna i minimalna temperatura, datum i vrijeme u ISO formatu, energetska efikasnost, propusnost punjenja, propusnost energije, trajanje punjenja, integracija vrijeme-temperatura, pauze, napon, unutrašnji otpor, struja i tip koraka. Svi ovi podaci predstavljaju rezultate mjerenja različitih baterija pod različitim uslovima, zbog čega su podaci raznovrsni. Na Slici 4.4 prikazan je grafikon promjene temperature baterije tokom ciklusa. Takođe, na Slici 4.5 nalazi se grafikon kapaciteta baterije tokom ciklusa pražnjenja.




*Slika 4.4. Grafikon promjene temperature baterije tokom ciklusa baterije pod nazivom FastCharge_000070_CH46_structure.json*




*Slika 4.5. Grafikon pražnjenja baterije iz Toyota skupa podataka tokom ciklusa baterije pod nazivom FastCharge_000070_CH46_structure.json*

### 4.2.1. Pretprocesiranje podataka

Podaci su podjeljeni na isti način kao podaci iz NASA skupa podataka što je opisano u odjeljku 4.1.1. Nakon što su podaci o baterijama učitani, bilo je potrebno izdvojiti relevantne informacije prikazane na Slici 4.3. Pošto nisu sve kolone od značaja za analizu, podaci su morali biti profiltrirani da bi se osigurala njihova upotrebljivost za dalja istraživanja.

Proces pretprocesiranja podataka o baterijama započinje učitavanjem sirovih podataka iz JSON datoteka koje sadrže informacije o baterijama. Prvi korak u ovom procesu uključuje organizovanje osnovnih informacija o baterijama, kao što su nazivi datoteka i indeksi ciklusa (i drugih podataka koji su prikazani na Slici 4.3), u strukturisan format, kako bi se obezbijedilo da svaka baterija bude jasno identifikovana kroz svoj životni ciklus.

Konkretno, podaci koji su korišteni uključuju: indeks, naziv datoteke baterije, kapacitet pražnjenja i punjenja, energiju pražnjenja i punjenja, unutrašnji otpor pri pražnjenju, kao i maksimalnu, prosečnu i minimalnu temperaturu. Dodatno, prikupljeni su podaci o energetskoj efikasnosti, ukupnom kapacitetu punjenja, ukupnoj energiji, trajanju punjenja, integrisanom vremenu temperature, statusu pauziranja, i datumu i vremenu mjerenja. Ovi podaci čine osnovu za dalje analize i omogućavaju detaljno praćenje performansi i stanja baterije tokom njenog životnog ciklusa.

Nakon inicijalnog strukturiranja podataka, fokus se prebacuje na detaljniju obradu različitih tipova podataka. Podaci se analiziraju ciklus po ciklusu, gdje se za svaki ciklus izdvajaju relevantni metrički podaci, poput kapaciteta pražnjenja. Ovaj pristup omogućava preciznije praćenje performansi baterije tokom vremena.

Vremenski podaci se zatim konvertuju iz ISO formata u UNIX vremenski pečat. Ovaj korak je važan jer omogućava lakše poređenje i manipulaciju vremenskim podacima, obezbeđujući da se svi događaji vezani za baterije mogu precizno pratiti u vremenskom kontekstu. Nakon konverzije, nepotrebne kolone, poput originalnih indeksa ciklusa i vremenskih podataka u ISO formatu, se uklanjaju da bi se smanjila složenost skupa podataka i obezbedilo da svi preostali podaci budu spremni za dalju analizu.

Tokom ovog procesa, prazni redovi se odbacuju kako bi podaci bili čisti i konzistentni. Očišćeni podaci su tada organizovani u strukturu koja omogućava lako pretraživanje i efikasnu manipulaciju, čime se pripremaju za primjenu mašinskog učenja.

Za primjenu mašinskog učenja na ovaj problem, neophodno je definisati ciljnu promjenljivu (eng. *target variable*) koja je korištena u analizi. U ovom eksperimentu, kao ciljna promjenljiva odabrana je SOH promjenljiva. U (2.1) prikazan je postupak izračunavanja ciljne promjenljive na osnovu kapaciteta baterije koji je korišten u pretprocesiranju. Vrijednost 1,1 Ah je uzeta kao referentni maksimalni kapacitet. Budući da su prisutni određeni *outlier*-i[9] koji uzrokuju da kapacitet blago prelazi ovu granicu oni predstavljaju anomalije i njihova vrijednost je ograničena na 1,1 Ah.

[9] izolovana vrijednost

# 5. Eksperimentalni rezultati

U izvedenim eksperimentima korištena je grid pretraga, implementirana kao *GridSearchCV* iz *scikit-learn Python* biblioteke. Grid pretraga omogućava pronalaženje najboljih hiperparametara na validacionom skupu podataka. U radnom okruženju, koje je kreirano kao polazna tačka eksperimenata, izvedene su varijacije ne samo hiperparametara, već i različitih algoritama. Često, rezultati na trening i validacionim skupovima mogu dovesti do preprilagođavanja, pa je u ovom slučaju predikcija izvršena nad testnim skupom. Vodeći se principom da ako se postignu najbolji rezultati na testnom skupu, koji nije bio uključen u trening proces, može se zaključiti da je preprilagođavanje izbjegnuto u okviru tog skupa podataka. Međutim, izazovi se mogu pojaviti pri upotrebi drugih skupova podataka, jer može doći do preprilagođavanja na skupu koji se koristi za treniranje.

Eksperimenti su sprovedeni na NASA i *Toyota* skupovima podataka, koji su opisani u poglavlju 4. U eksperimentima su podaci podijeljeni na trening i testne skupove, pri čemu su trening podaci korišteni i za validaciju putem tehnike unakrsne validacije (eng. *cross-validation*). Unakrsna validacija je tehnika koja se koristi u mašinskom učenju i statistici za procjenu sposobnosti modela da generalizuje na nove, neviđene podatke. Ovo je efikasan način za provjeru koliko je model "dobar" i koliko će biti efektivan kada se primijeni na stvarne podatke koji nisu korišteni tokom treniranja.

Eksperimenti su obuhvatili primjenu regresije s ciljem predviđanja SOH-a kao ciljne promjenljive, kao i klasifikaciju zasnovanu na RUL metrici kao ciljnoj promjenljivoj. Pored toga, sproveden je i eksperiment s prijenosom učenja, pri čemu je korišten regresivni model na ulazu, dok se na izlazu tražila kategorička ciljna promjenljiva izvedena iz RUL metrike.Ulazne promjenljive su bili podaci iz NASA i *Toyota* skupova podataka, koji su prethodno prošli kroz proces pretprocesiranja. Detalji ovog procesa za NASA skup podataka navedeni su u odjeljku 4.1.1, dok su postupci pretprocesiranja za *Toyota* skup podataka također opisani u odjeljku 4.2.1. Eksperimenti su uključivali varijacije u procentu podataka korištenih za trening, mijenjajući obim trening skupa sa 100% na 30%. Ova varijacija je omogućila procjenu sposobnosti algoritma da prepozna šablone u podacima, bez da dođe do učenja napamet.

U kontekstu klasifikacije, ispitivani su svi ciklusi, kao i samo prvih 50 ciklusa, uzimajući u obzir da kasniji ciklusi možda nemaju značajan uticaj na predviđanje RUL metrike. Ovaj pristup omogućio je procjenu uticaja različitih segmenata ciklusa na preciznost modela klasifikacije.

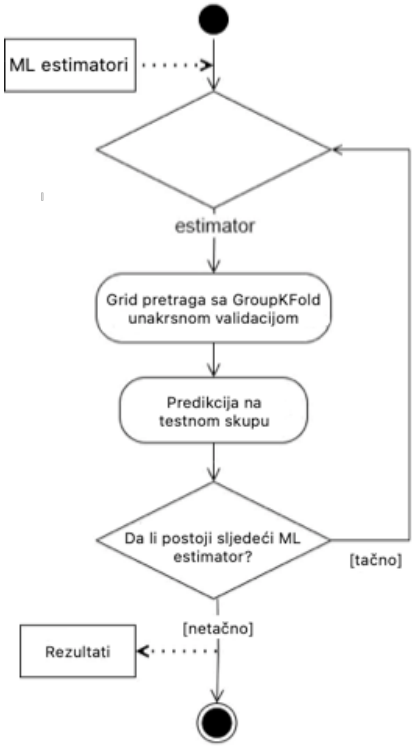
Svi eksperimenti dijele isti radni okvir za pripremu podataka i treniranje modela. Proces uključuje sveobuhvatan pristup koristeći široko prepoznate industrijske alate i tehnike.

Za pronalaženje optimalnih parametara za svaki algoritam mašinskog učenja korišten je *GridSearchCV* iz *scikit-learn* biblioteke. Validacija je sprovedena pomoću *GroupKFold* unakrsne validacije, takođe iz *scikit-learn biblioteke*, koja osigurava da nema preklapanja između podataka za obuku i testiranje, čime se efikasno vrši prevencija prekomjernog učenja modela. Za modele neuronskih mreža korištena je *TensorFlow* biblioteka, dok su za integraciju tih modela u grid pretragu korišteni *KerasRegressor i KerasClassifier* iz scikeras biblioteke.

Modeli neuronskih mreža trenirani su tokom 50 epoha sa veličinom serije od 128, uzimajući u obzir relativno mali obim skupa podataka.

Eksperimenti sa NASA skupom podataka sprovedeni su na Lenovo Yoga Slim 7 laptopu, opremljenom Ryzen 7 4800U procesorom sa x86 arhitekturom, 16GB RAM-a i integrisanom AMD Radeon RX Vega 8 grafičkom karticom. Eksperimenti sa Toyota skupom podataka, koji je znatno veći, sprovedeni su na Apple MacBook Pro laptopu sa Apple M1 Pro čipom sa ARM arhitekturom, 32GB RAM-a i integrisanom grafičkom karticom. Promjena okruženja bila je neophodna zbog pregrijavanja *Lenovo* laptopa tokom dugotrajnih grid pretraga, što je rezultiralo čestim prisilnim gašenjem računara.

Proces grid pretrage započinje kreiranjem *pipeline*-a koji uključuje skaler i estimator za svaki algoritam mašinskog učenja. Estimator predstavlja dio *pipeline*-a koji prima prethodno obrađene podatke (koji su, na primer, prošli kroz proces skaliranja pomoću skalera) i zatim se vrši obučavanje na tim podacima kako bi se optimizovali njegovi hiperparametri. Cilj grid pretrage je da pronađe optimalne vrednosti hiperparametara za odabrani estimator kako bi se postigao što bolji učinak modela. Ovaj proces je ilustrovan pomoću UML dijagrama aktivnosti prikazanog na Slici 5.1.




*Slika 5.1. Proces grid pretrage - UML dijagram aktivnosti*

Optimalni hiperparametri se određuju putem unakrsne validacije na trening skupu, nakon čega se model testira na zasebnom testnom skupu. Regresivni modeli se evaluiraju koristeći metrike kao što su MSE i R-kvadrat u odnosu na stvarne vrijednosti SOH. Modeli se rangiraju prema njihovim MSE rezultatima, a najbolji algoritam na testnom skupu se identifikuje i njegova podešavanja se čuvaju za buduću upotrebu.

Za klasifikacione modele, glavna metrika za evaluaciju je tačnost. Pored toga, izračunavaju se i druge metrike kao što su F1, preciznost, odziv i matrica konfuzije kako bi se dobila sveobuhvatna ocjena modela na testnom skupu.

U eksperimentima ovog rada, ključne metrike na osnovu kojih se modeli porede su MSE za regresiju i tačnost za klasifikaciju.

# 5.1. Primjena regresije sa NASA skupom podataka

U istraživanju je fokus bio na procjeni tačnosti različitih algoritama mašinskog učenja u predikciji stanja zdravlja (SOH) litijum-jonskih baterija na NASA skupu podataka. U eksperimentu korišteni su algoritmi iz grupe klasičnih algoritama mašinskog učenja i algoritmi dubokog učenja. Iz grupe klasičnih algoritama su korišteni SVM, linearna regresija, *Ridge*, *Lasso*, RF, i algoritmi *gradient boosting*-a kao što su *XGBoost* i *CatBoost*. Iz grupe dubokog učenja korištene su MLP, CNN i LSTM neuronske mreže.

Za optimizaciju hiperparametara svakog modela korištena je grid pretraga u kombinaciji sa unakrsnom validacijom na trening skupu. U ovom procesu, korištena je *GroupKFold* unakrsna validacija sa 5 foldova, što omogućava bolju procjenu performansi modela kroz različite grupe podataka. *GroupKFold* unakrsna validacija osigurava da svi uzorci iz iste grupe budu zajedno u istom foldu, sprečavajući curenje podataka između trening i validacionih skupova, čime se obezbjeđuje realnija procjena performansi modela. tj. smanjuje se šansa za *overfitting*-om. Nakon toga, modeli su testirani na posebnom testnom skupu, a rezultati su prikazani u Tabeli 5.1.

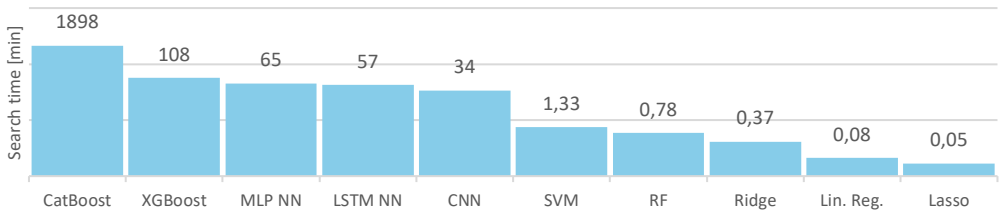
*Tabela 5.1. Rezultati primjene regresije sa NASA skupom podataka*

| | ML algoritam | MSE | R2 |
|---|---|---|---|
| **1.** | MLP NN | 0.0012 | 0.9716 |
| **2.** | CNN | 0.0032 | 0.9240 |
| **3.** | *CatBoost* | 0.0037 | 0.9142 |
| **4.** | *XGBoost* | 0.0042 | 0.9014 |
| **5.** | *Random forest* | 0.0054 | 0.8727 |
| **6.** | *Lasso* | 0.0072 | 0.8312 |
| **7.** | *Ridge* | 0.0081 | 0.8107 |
| **8.** | *Linear regression* | 0.0135 | 0.6821 |
| **9.** | LSTM NN | 0.0214 | 0.4989 |
| **10.** | SVM | 0.0237 | 0.4444 |

Rezultati su pokazali da je MLP neuronska mreža bila najtačnija, sa najnižom srednjom kvadratnom greškom (MSE) od 0,0012 i najvišim koeficijentom determinacije ($R^2$) od 0,9716, što ukazuje na visoku tačnost modela. CNN i napredni ansambl modeli, poput *CatBoost*-a i *XGBoost*-a, takođe su ostvarili dobre rezultate, ali sa nešto slabijim performansama u poređenju sa MLP modelom.

Vrijeme potrebno za *grid search* je pod uticajem složenosti modela i broja hiperparametara, kao što je prikazano na Slici 5.2. Ovi rezultati naglašavaju potencijal

prilagođenih mašinskih modela za praćenje zdravlja baterija, pri čemu se MLP mreža ističe kao najefikasnija. Iako nisu dostigli nivoe tačnosti vodećih modela, RF, Lasso i Ridge regresija su pokazali solidne performanse, što ih čini korisnim za osnovne analize ili dalja poboljšanja u budućim studijama. Ovi modeli ističu raznolikost pristupa i značaj pažljivog podešavanja hiperparametara.

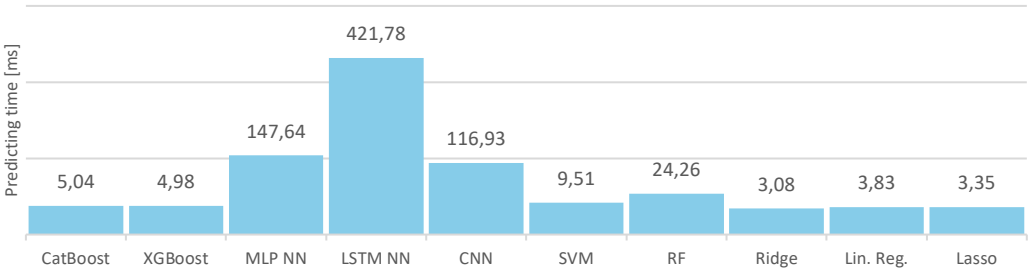



*Slika 5.2. Trajanje grid search-a modela*

Uočeno je da modeli kao što su *CatBoost* i *XGBoost* zahtijevaju duže vrijeme za pretragu optimalnih hiperparametara zbog složenosti i velikog broja parametara. S druge strane, modeli sa manje hiperparametara, kao što su SVM, RF i linearni modeli, imaju kraće vrijeme pretrage. *CatBoost* je pokazao najduže vreme pretrage, dok su *XGBoost*, MLP i LSTM takođe zahtijevali više vremena zbog složenijih struktura. CNN je bio brži, vjerovatno zbog efikasnije arhitekture, dok su linearni modeli bili među najbržima, zbog manje složenosti u računanjima i manjem broju parametara za podešavanje.

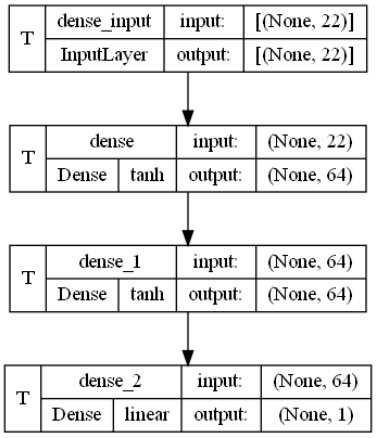
Razumijevanje vremena potrebnog za optimizaciju hiperparametara je ključno za upravljanje resursima i planiranjem vremenskih rokova u projektima mašinskog učenja. To može značajno uticati na izbor algoritma, posebno u situacijama gde su potrebni brzi razvojni ciklusi.

Podaci iz Tabele 5.1 i Slike 5.3 ukazuju na kompromis između preciznosti modela i vremena predikcije. MLP i CNN modeli pokazuju duže vrijeme predikcije. U poređenju, modeli kao što su *CatBoost* i *XGBoost* postižu bolji balans između brzine predikcije i tačnosti.




*Slika 5.3. Trajanje predikcije modela*

Arhitektura MLP modela, prikazana na Slici 5.4, rezultat je detaljne optimizacije hiperparametara korištenjem grid pretrage.



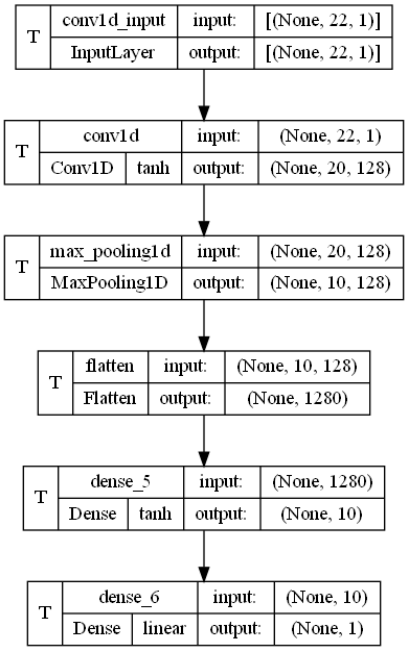

*Slika 5.4. Arhitektura MLP modela*

Tabela 5.2 prikazuje detaljan pregled parametara pretrage i optimalnih hiperparametara za MLP model, utvrđenih kroz *grid search* optimizaciju.

*Tabela 5.2. Prostor pretraživanja i optimalne vrijednosti hiperparametara za MLP model*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | true |
| mlp-nn__model__neurons_layer_1 | [32, 64, 128] | 64 |
| mlp-nn__model__neurons_layer_2 | [16, 32, 64] | 64 |
| mlp-nn__model__activation | ["relu", "tanh", "sigmoid"] | "tanh" |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | 100 |

Prikazana arhitektura CNN modela na Slici 5.5 je optimizovana za predikciju SOH-a litijum-jonskih baterija, sa detaljima u Tabeli 5.3 koji pokazuju optimalne vrijednosti hiperparametara.




*Slika 5.5. Arhitektura CNN modela*

Tabele 5.4 i 5.5 pružaju pregled optimalnih hiperparametara za *CatBoost* i *XGBoost* algoritme, naglašavajući važnost preciznog podešavanja parametara za poboljšanje performansi modela. Ove tabele ističu ključne parametre koji su se pokazali najefikasnijim u predikciji SOH-a litijum-jonskih baterija.

*Tabela 5.3. Prostor pretraživanja i optimalne vrijednosti hiperparametara kod CNN modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | false |
| cnn-nn__model__filters | [32, 64, 128] | 128 |
| cnn-nn__model__kernel_size | [2, 3, 5] | 3 |
| cnn-nn__model__dense_units | [10, 20, 50] | 10 |
| cnn-nn__model__activation | ["relu", "tanh", "sigmoid"] | "tanh" |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | "adam" |

Tabela 5.5 prikazuje sličnu optimizaciju za *XGBoost* algoritam, ponovo predstavljajući najefikasnije vrijednosti hiperparametara nakon testiranja različitih mogućnosti. Odabrani parametri odražavaju one koji doprinose najboljoj prediktivnoj snazi modela prema rezultatima grid pretrage.

*Tabela 5.4. Prostor pretraživanja i optimalni hiperparametri kod CatBoost modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | false |
| catboost__depth | [4, 5, 6] | 4 |
| catboost__learning_rate | [0.01, 0.1, 0.2] | 0.1 |
| catboost__l2_leaf_reg | [0.1, 0.2, 0.3] | 0.1 |
| catboost__min_child_weight | [100, 200, 300] | 100 |
| catboost__subsample | [0.5, 0.75, 1] | 0.75 |
| catboost__colsample_bylevel | [0.5, 0.75, 1] | 0.75 |
| catboost__loss_function | ["RMSE", "MAE", "Quantile:alpha=0.5"] | “RMSE” |
| catboost__bootstrap_type | ["Bayesian", "Bernoulli", "MVS"] | "Bernoulli" |

*Tabela 5.5. Prostor pretraživanja i optimalni hiperparametri kod XGBoost modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | true |
| xgboost__n_estimators | [50, 100, 300] | 300 |
| xgboost__max_depth | [4, 5, 6] | 6 |
| xgboost__learning_rate | [0.01, 0.1, 0.2] | 0.1 |
| xgboost__min_child_weight | [1, 2, 3] | 2 |
| xgboost__gamma | [0, 0.1, 0.2, 0.3] | 0 |
| xgboost__subsample | [0.5, 0.75, 1] | 0.75 |
| xgboost__colsample_bytree | [0.5, 0.75, 1] | 0.5 |
| xgboost__reg_alpha | [0, 0.1, 0.5] | 0 |
| xgboost__reg_lambda | [0, 0.1, 0.5] | 0.5 |

Obje tabele zajedno ističu uticaj pažljivog podešavanja hiperparametara na performanse modela, prikazujući rafinisana podešavanja koja su dala najperspektivnije rezultate za svaki algoritam u predikciji stanja zdravlja (SOH) litijum-jonskih baterija.

## 5.2. Efekat vremenske informacije na predikciju unutar ciklusa

U eksperimentu opisanom u tački 5.1 korišteni su agregirani podaci dobijeni tokom perioda pražnjenja baterija unutar ciklusa. Cilj ovog eksperimenta bio je da se postignu bolji rezultati treniranja uzimajući u obzir izmjerene podatke koji uključuju informacije o pražnjenju baterija.

Razmatrana su dva potencijalna pristupa: korištenje rekurentne neuronske mreže (RNN) i konvolucione neuronske mreže (CNN). Zbog suboptimalnih rezultata postignutih u prethodnom eksperimentu sa rekurentnom neuronskom mrežom, kao i zbog strukture skupa podataka, u ovom eksperimentu je odlučeno da se koristi isključivo konvoluciona neuronska mreža.

NASA skup podataka ima tri dimenzije: baterija, ciklus i perioda. Sve tri dimenzije su uzete u obzir prilikom modelovanja CNN-a. Iako se na osnovu toga moglo očekivati korištenje 3D konvolucione mreže, 2D konvolucija se pokazala kao prikladnije rješenje. U 2D konvolucionoj mreži, odgovarajući Conv2D sloj (iz *Tensorflow* biblioteke) prima 4D tenzor kao ulazni argument [80]:

*(batch_size, new_height, new_width, filters)*

Veličina *batch*-a u ovom slučaju predstavlja broj baterija, nova visina predstavlja broj ciklusa, nova širina predstavlja broj perioda po ciklus, dok filteri predstavljaju izlaznu dimenziju sloja. Takođe je važno navesti jezgro (eng. *kernel*) koje je korišteno za konvoluciju unutar više ciklusa i više perioda. Odabrano je jezgro veličine $3 \times 3$.

### 5.2.1. Izazovi

Prvi izazov bio je osigurati odgovarajuću strukturu podataka pogodnu za 2D konvolucioni sloj. U eksperimentu 5.1 korištena je struktura *DataFrame* iz *Pandas* biblioteke. Međutim, u trenutku izvođenja eksperimenta, nije postojala mogućnost rada sa *DataFrame*-om koji ima 4 dimenzije, što je bilo neophodno jer 2D konvolucioni sloj zahtijeva 4D tenzor. Stoga je bilo potrebno konstruisati 4D niz koristeći funkcionalnosti *numpy* biblioteke. Kako bi se postigle optimalne performanse i kvalitet koda, primijenjena je kombinacija funkcija *stack* i *expand_dims* iz *numpy* biblioteke.

Drugi izazov bio je osigurati adekvatnu strukturu podataka, budući da je bilo neophodno da broj perioda unutar svakog ciklusa bude ujednačen kako bi se mogao formirati trodimenzionalni niz. Da bi se ovaj problem prevazišao, primijenjena je tehnika interpolacije, a u specifičnim slučajevima i ekstrapolacije. Prvi korak je bio identifikacija maksimalnog broja perioda unutar pojedinačnog ciklusa, nakon čega su svi ostali ciklusi dopunjeni interpolacijom kako bi se postigla uniformnost u broju elemenata. Implementacija ovog pristupa detaljno je opisana u nastavku. Funkcija za interpolaciju dizajnirana je da prilagodi dužinu niza na unaprijed zadatu vrijednost. Ako je ulazni niz već u željenom formatu, funkcija samo kreira raspored tačaka koji pokriva opseg niza. Potom se generiše interpolaciona funkcija koja koristi

linearni metod interpolacije. Ova funkcija omogućava interpolaciju vrijednosti unutar niza, kao i ekstrapolaciju izvan originalnog opsega podataka. Nakon toga, funkcija kreira novi raspored tačaka, pokrivajući isti opseg. Interpolaciona funkcija se primjenjuje na nove tačke kako bi se dobio novi niz sa interpoliranim vrijednostima, čija je dužina jednaka zadatoj vrijednosti. Ako niz nije u željenom formatu, funkcija jednostavno vraća niz sa ponovljenim vrijednostima kako bi se postigla potrebna dužina.

Treći izazov se desio nakon kreiranja podataka u toku pretprocesiranja. Kao i u prethodnim eksperimentima, korišten je pristup koji je uključivao skaliranje podataka. Problem je nastao prilikom skaliranja podataka, jer metoda korištena za skaliranje podržava samo jednimenzionalne i dvodimenzionalne podatke, dok su podaci bili u višedimenzionalnom formatu. Da bi se ovaj problem prevazišao, implementiran je pristup koji omogućava skaliranje podataka različitih dimenzija kroz preoblikovanje podataka u dvodimenzionalni oblik prije skaliranja i vraćanje podataka u originalni oblik nakon skaliranja. Ovaj pristup osigurava da broj elemenata ostane isti, čime se izbjegavaju problemi sa dimenzijama. Koristeći ovu metodu, podaci se prvo preoblikuju, zatim skaliraju, i konačno vraćaju u originalni oblik, omogućavajući pravilno skaliranje višedimenzionalnih podataka bez gubitka informacija.

### 5.2.2. Rezultati eksperimenta

Rezultati su dati u Tabeli 5.6. Iako je srednja kvadratna greška mala, ne predstavlja poboljšanje u odnosu na rezultate inicijalnog eksperimenta opisanog u poglavlju 5.1. Ipak, ostvareno je poboljšanje u odnosu na rezultat dobijen pomoću CNN-a. Pretpostavka je da su periodi različitih ciklusa dosta različiti što se može agregacijom izuzeti pri čemu se onda može doći do boljeg učenja i predikcije.

*Tabela 5.6. Rezultati efekta vremenske informacije na predikciju*

| Trajanje treniranja | Trajanje predikcije | MSE | R2 | Najbolji grid parametri | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | batch_size | epochs | activation | dense_units | filters | kernel_size | optimizer | with_std |
| 4293 | 0.828 | 0.023 | 0.611 | 128 | 50 | Relu | 20 | 128 | [3,3] | rmsprop | true |

Takođe NASA skup podataka nema puno podataka, što može biti jedan od razloga zašto ovaj eksperiment ne donosi veliko poboljšanje, s obzirom da neuronske mreže rade mnogo bolje kad imaju mnogo podataka, a ovo nije bio slučaj.

## 5.3. Primjena regresije sa *Toyota* skupom podataka

U ovom eksperimentu, glavni cilj bio je prikazati performanse algoritama mašinskog učenja koristeći *Toyota* skup podataka i SOH metriku. Detalji vezani za učitavanje, obradu i pretprocesiranje skupa podataka opisani su u poglavlju 4.2. Za algoritme koji su ostvarili najbolje performanse u eksperimentu 5.1 sprovedena je grid pretraga parametara. Ti algoritmi su *XGBoost* i *CatBoost*, kao predstavnici tradicionalnih algoritama mašinskog učenja, te MLP i CNN, kao predstavnici algoritama dubokog učenja (neuronskih mreža).

Format podataka koji je korišten u grid pretrazi prikazan je na Slici 5.6. Podaci prikazani na Slici 4.3, a koji nisu prisutni na Slici 5.6, uklonjeni su zbog toga što su sadržavali NaN vrijednosti.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| battery_filename | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... | /Users/... |
| charge_capacity | 1.44133... | 1.06622... | 1.07023... | 1.07149... | 1.07223... | 1.07308... | 1.07324... | 1.0734178 | 1.07379... | 1.07379... | 1.07388... |
| discharge_energy | 6.18299... | 3.24487... | 3.26163... | 3.26379... | 3.26565... | 3.27067... | 3.26870... | 3.26820... | 3.27126... | 3.26961... | 3.270534 |
| charge_energy | 4.75505... | 3.71438... | 3.72089... | 3.72342... | 3.72621... | 3.72811... | 3.72830... | 3.72956... | 3.72998... | 3.73012... | 3.73096... |
| dc_internal_resistance | 0.02938... | 0.01735... | 0.01715... | 0.01720... | 0.01712... | 0.01701... | 0.01710... | 0.01709... | 0.01698... | 0.01707... | 0.01705... |
| temperature_maximum | 33.9890... | 36.4375... | 36.9092... | 36.9016... | 36.8171... | 36.7804... | 36.7056... | 36.6531... | 36.7110... | 36.6183... | 36.7859... |
| temperature_average | 30.0545... | 32.3680... | 32.7670... | 32.7306... | 32.6528... | 32.8553... | 32.5789... | 32.4239... | 32.5800... | 32.4081... | 32.4450... |
| temperature_minimum | 27.8666... | 30.0639... | 30.4147... | 30.1320... | 30.3244... | 30.4732... | 29.9752... | 29.9511... | 30.0200... | 29.8075... | 29.9891... |
| energy_efficiency | 1.30030... | 0.87359... | 0.87657... | 0.87655... | 0.87640... | 0.87730... | 0.87672... | 0.87629... | 0.87701... | 0.87654... | 0.87659... |
| charge_throughput | 1.44133... | 2.50755... | 3.57779... | 4.64928... | 5.72151... | 6.79460... | 7.86784... | 8.94126... | 10.0150... | 11.0888... | 12.1627... |
| energy_throughput | 4.75505... | 8.46944... | 12.1903... | 15.9137... | 19.6399... | 23.3680... | 27.0963... | 30.8259... | 34.5559... | 38.2860... | 42.0170... |
| charge_duration | 33024.0 | 640.0 | 640.0 | 640.0 | 640.0 | 640.0 | 512.0 | 512.0 | 512.0 | 640.0 | 640.0 |
| time_temperature_integrated | 38666.0... | 1933.53... | 1953.84... | 1951.37... | 1948.51... | 2028.56... | 1943.46... | 1936.19... | 1945.13... | 1933.16... | 1935.19... |
| paused | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| timestamp | 1498881... | 149895... | 149896... | 149896... | 149896... | 149897... | 1498977... | 149898... | 149898... | 149898... | 1498991... |

*Slika 5.6. Pripremljen dataframe sa podacima (transponovan dataframe)*

Kao što je navedeno u poglavlju 5, svi eksperimenti sa regresijom su sprovedeni dva puta: prvi put sa cijelim skupom podataka, a drugi put koristeći 30% skupa podataka.

U Tabeli 5.7 prikazani su rezultati eksperimenta. *CatBoost* je ostvario najbolje rezultate, sa izuzetno niskom srednjom kvadratnom greškom od $8.4 \times 10^{-7}$ i visokom $R^2$ metrikom od 0.9995. Na 30% skupa podataka, *CatBoost* je takođe nadmašio ostale algoritme, iako je greška bila nešto veća u odnosu na inicijalno mjerenje. Konvoluciona neuronska mreža je pokazala izvrsne rezultate, slično kao i *XGBoost*, dok je potpuno povezana neuronska mreža imala nešto slabije performanse. Upoređujući rezultate na manjem skupu podataka s rezultatima na cijelom skupu, uočava se da neuronske mreže zahtijevaju veći obim podataka kako bi nadmašile klasične algoritme mašinskog učenja. U ovom slučaju, *XGBoost* je preuzeo prednost nad konvolucionom neuronskom mrežom kada se koristi manji skup podataka za obučavanje modela.

Što se tiče trajanja grid pretrage, *CatBoost* je značajno zahtijevao više vremena u poređenju s ostalim algoritmima. Grid pretraga za *CatBoost* trajala je 210 234 sekunde, što je ekvivalentno nešto više od 2 dana i 10 sati, što predstavlja izuzetno dug period. S druge strane, pretraga optimalnih hiperparametara za konvolucionu neuronsku mrežu trajala je nešto manje od 14 sati, za *XGBoost* algoritam nešto manje od 5 sati, dok je za potpuno povezanu mrežu trajala oko 4,5 sata.

*Tabela 5.7. Rezultati regresije nad cijelim Toyota skupom podataka*

| Rang | Algoritam | Trajanje grid pretrage[10] | Trajanje predikcije | | MSE | | R2 | |
|---|---|---|---|---|---|---|---|---|
| | | 100% | 100% | 30% | 100% | 30% | 100% | 30% |
| **1.** | ***CatBoost*** | 210234 | 0.0095 | 0.0106 | **8.4e-07** | **1.75e-06** | **0.9995** | **0.9990** |
| 2. | CNN | 50052 | 0.3952 | 0.3111 | 9.3e-07 | 2.12e-05 | 0.9994 | 0.9875 |
| 3. | *XGBoost* | 18223 | **0.0079** | **0.0093** | 9.7e-07 | 5.09e-06 | 0.9994 | 0.9870 |

[10] Grid pretraga nije izvršavana za 30% skupa podataka; umjesto toga, direktno su korišteni hiperparametri prethodno optimizovani tokom grid pretrage na cijelom skupu podataka, kako bi se uštedjelo vrijeme.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4. | MLP | **16301** | 0.4110 | 0.3099 | 1.78e-06 | 5.5e-06 | 0.9989 | 0.9967 |

U Tabelama 5.8, 5.9, 5.10 i 5.11 prikazane su oblasti pretraživanja sa vrijednostima optimalnih hiperparametara za grid pretrage za sve prethodno navedene algoritme iz Tabele 5.7.

*Tabela 5.8. Prostor pretraživanja i optimalne vrijednosti parametara CatBoost modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | false |
| catboost__depth | [4, 5, 6] | 4 |
| catboost__learning_rate | [0.01, 0.1, 0.2] | 0.2 |
| catboost__l2_leaf_reg | [0.1, 0.2, 0.3] | 0.2 |
| catboost__min_child_weight | [100, 200, 300] | 100 |
| catboost__subsample | [0.5, 0.75, 1] | 0.5 |
| catboost__colsample_bylevel | [0.5, 0.75, 1] | 0.75 |
| catboost__loss_function | ["RMSE", "MAE", "Quantile:alpha=0.5"] | “RMSE” |
| catboost__bootstrap_type | ["Bayesian", "Bernoulli", "MVS"] | "MVS" |

*Tabela 5.9. Prostor pretraživanja i optimalne vrijednosti parametara CNN modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | false |
| cnn-nn__model__filters | [32, 64, 128] | 128 |
| cnn-nn__model__kernel_size | [2, 3, 5] | 3 |
| cnn-nn__model__dense_units | [10, 20, 50] | 10 |
| cnn-nn__model__activation | ["relu", "tanh", "sigmoid"] | "tanh" |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | "adam" |

*Tabela 5.10. Prostor pretraživanja i optimalne vrijednosti parametara XGBoost modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | true |
| xgboost__n_estimators | [50, 100, 300] | 100 |
| xgboost__max_depth | [4, 5, 6] | 4 |
| xgboost__learning_rate | [0.01, 0.1, 0.2] | 0.1 |
| xgboost__min_child_weight | [1, 2, 3] | 1 |
| xgboost__gamma | [0, 0.1, 0.2, 0.3] | 0 |
| xgboost__subsample | [0.5, 0.75, 1] | 1 |
| xgboost__colsample_bytree | [0.5, 0.75, 1] | 1 |
| xgboost__reg_alpha | [0, 0.1, 0.5] | 0.1 |
| xgboost__reg_lambda | [0, 0.1, 0.5] | 0.5 |

*Tabela 5.11. Prostor pretraživanja i optimalne vrijednosti parametara MLP modela*

| | **Oblast pretraživanja** | **Optimalni hiper parametri** |
|---|---|---|
| scaller__with_std | [true, false] | true |
| mlp-nn__model__neurons_layer_1 | [32, 64, 128] | 128 |
| mlp-nn__model__neurons_layer_2 | [16, 32, 64] | 16 |
| mlp-nn__model__activation | ["relu", "tanh", "sigmoid"] | "sigmoid" |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | “adam” |

Na Slikama 5.4 i 5.5 su prikazane arhitekture potpuno povezane i konvolucione neuronske mreže koje su se takođe koristile i u ovom eksperimentu.

## 5.4. Primjena klasifikacije sa *Toyota* skupom podataka

Predikcija zdravlja baterije korištenjem RUL metrike i regresije ne daje dobre rezultate, jer je veoma teško da algoritam precizno odredi RUL vrijednost. Glavni razlog za to je što su promjene u svakom redu trening skupa minimalne, dok su promjene u vrijednostima RUL metrike daleko veće (cjelobrojne). Pretpostavka je bila da bi grupisanje tih RUL vrijednosti u kategorije moglo pozitivno uticati na predikciju zdravlja baterije. Kada se ciljne promjenljive grupišu, umjesto regresije koristi se klasifikacija.

Tokom eksperimenta uočen je problem s velikim brojem odstupajućih vrijednosti u podacima, pa je bilo neophodno izvršiti njihovu normalizaciju. U ovom postupku se obrađuje kolona koja sadrži vrijednosti kapaciteta pražnjenja baterije s ciljem identifikacije i zamjene odstupajućih vrijednosti. Prvo se podaci konvertuju u numerički format, a zatim se za svaku vrijednost računa pokretna medijana koja uzima u obzir trenutnu i prethodne dvije vrijednosti. Razlika između pokretne medijane i stvarne vrijednosti koristi se za prepoznavanje odstupanja. Ako odstupanje prelazi određeni prag, stvarna vrijednost se zamjenjuje pokretnom medijanom (prozora veličine 3), što pomaže u normalizaciji podataka i smanjenju uticaja ekstremnih vrijednosti što je posebno efikasno kod vremenskih serija i drugih kontinuiranih podataka, gdje su nagla odstupanja često rezultat grešaka ili vanrednih događaja.

Nakon normalizacije kapaciteta pražnjenja, važno je izračunati metriku preostalog vijeka trajanja (RUL) baterija, koja pruža uvid u to koliko ciklusa pražnjenja baterija može izdržati prije nego što postane neupotrebljiva. Proces započinje određivanjem praga kapaciteta ispod kojeg se smatra da je baterija značajno degradirana. U ovoj analizi, prag je postavljen na 80%, jer većina baterija u skupu podataka pokazuje da su u dobrom stanju. Ovaj prag omogućava precizno praćenje trenutka kada baterija počinje gubiti svoj kapacitet na upotrebljiv način, identifikujući samo one baterije koje pokazuju prve znakove degradacije. Za svaku bateriju se prati ciklus pražnjenja, te se određuje prvi trenutak kada kapacitet padne ispod ovog definisanog praga. Na osnovu ove informacije, izračunava se koliko još ciklusa baterija može izdržati prije nego što njen vijek trajanja završi, omogućavajući preciznije predviđanje zamjene baterija.

Podaci o baterijama se organizuju prema jedinstvenim identifikatorima, što omogućava detaljno praćenje stanja svake pojedinačne baterije. Ovakva organizacija podataka omogućava da se za svaku bateriju identifikuje tačka u kojoj je njen kapacitet počeo značajno opadati, pružajući ključne informacije o njenoj dugovječnosti. Kontinualnim praćenjem performansi baterija, moguće je identificirati znakove degradacije u ranoj fazi i predvidjeti trenutak kada će biti potrebna zamjena. Ovaj pristup ne samo da optimizuje performanse i pouzdanost sistema, već također omogućava pravovremeno planiranje zamjene baterija, čime se smanjuju rizici i održavaju optimalni uslovi rada.

Da bi se izvršila klasifikacija, potrebno je uvesti kategorije kao ciljne promjenljive. Odabrani opsezi RUL kategorija prikazani su u Tabeli 5.12. Pragovi su odabrani na osnovu analize izračunatih RUL vrijednosti i njihovih raspodjela.

*Tabela 5.12. Odabrani opsezi RUL kategorija u klasifikaciji*

| Opseg RUL vrijednosti | Klasifikaciona kategorija | Opis |
|---|---|---|
| 0 | *expired* | Baterija je potpuno degradirana i više se ne može koristiti. |
| 1-100 | *short_lifespan* | Baterija ima vrlo malo preostalih ciklusa i uskoro će biti potrebna zamjena. |
| 101-300 | *medium_lifespan* | Baterija ima umjeren broj preostalih ciklusa, još uvijek funkcionalna, ali se bliži kraju svog vijeka. |
| 301-500 | *long_lifespan* | Baterija je u dobrom stanju s mnogo preostalih ciklusa, pouzdana za korištenje još neko vrijeme. |
| 501 i više | *very_long_lifespan* | Baterija je u izvrsnom stanju s vrlo dugim preostalim vijekom trajanja, minimalni znakovi degradacije. |

U Tabeli 5.13 prikazani su rezultati eksperimenta. Analiza rezultata klasifikacije na *Toyota* skupu podataka otkrila je nekoliko ključnih zaključaka. *CatBoost* algoritam se istakao sa najvišom tačnošću od 0.8554 i najvišim F1 rezultatom od 0.7229 kada su korišteni svi ciklusi podataka, što ga čini najpouzdanijim izborom za preciznu klasifikaciju u ovom kontekstu. Također, *CatBoost* je pokazao najkraće vrijeme predikcije od 0.0098 sekundi za 50 ciklusa, što ga čini vrlo efikasnim za brze analize, uprkos najdužem trajanju grid pretrage.

*XGBoost* algoritam je ostvario vrlo visoku tačnost od 0.8495 i F1 rezultatom od 0.7228 za sve cikluse, što je vrlo blizu performansama *CatBoost*-a. *XGBoost* se također izdvojio kao najbrži u pogledu trajanja predikcije za sve cikluse, sa vremenom od 0.0271 sekundi, što ga čini efikasnim izborom u situacijama kada su resursi ograničeni.

CNN algoritam se posebno istakao po najvišoj tačnosti za prvih 50 ciklusa, sa tačnošću od 0.8171, ali ima znatno duže vrijeme predikcije, što može biti ograničavajuće za aplikacije koje zahtijevaju brzu obradu podataka. Njegova tačnost za sve cikluse iznosi 0.8485, što je vrlo konkurentno, ali produženo vrijeme predikcije ostaje izazov.

MLP algoritam pokazuje solidne performanse sa tačnošću od 0.8369 za sve cikluse i 0.7964 za 50 ciklusa, kao i F1 rezultatom od 0.7064 za sve cikluse, ali nije se istakao kao najbolji u bilo kojoj specifičnoj kategoriji. Njegovo duže vrijeme predikcije može biti faktor koji ga čini manje atraktivnim u poređenju sa *CatBoost*-om i *XGBoost*-om.

Slučajna šuma i SVM algoritmi, iako najbrži u pogledu trajanja grid pretrage, pokazuju niže vrijednosti tačnosti i F1 vrijednosti. Slučajna šuma postiže tačnost od 0.8346 i F1 vrijednost od 0.6473, dok SVM ima najnižu tačnost od 0.8140 i F1 vrijednost od 0.3991 za sve cikluse.

Iako postoje relativne razlike među algoritmima, svi prikazani algoritmi daju solidne rezultate u kontekstu ove studije. Sveukupno, *CatBoost* se izdvaja kao najefikasniji algoritam za rad sa cjelokupnim skupom podataka zbog svoje visoke tačnosti i F1 vrijednosti, dok se CNN pokazuje najboljim za manji broj ciklusa. *XGBoost* pruža odličan balans između performansi i efikasnosti, sa najboljim vremenom predikcije za sve cikluse. MLP, slučajna

šuma i SVM nude kompromis između vremena pretrage i tačnosti, ali nisu najbolji izbor za postizanje maksimalnih performansi.

Ovi rezultati omogućavaju donošenje informisanih odluka pri izboru algoritama za različite scenarije i aplikacije, uzimajući u obzir balans između tačnosti, vremena predikcije i trajanja grid pretrage.

*Tabela 5.13. Rezultati klasifikacije nad Toyota skupom podataka*

| Rang | Algoritam | Trajanje grid pretrage | | Trajanje predikcije | | Tačnost | | F1 | |
|---|---|---|---|---|---|---|---|---|---|
| | | Svi ciklusi | 50 ciklusa | Svi ciklusi | 50 ciklusa | Svi ciklusi | 50 ciklusa | Svi ciklusi | 50 ciklusa |
| **1.** | ***CatBoost*** | 287580 | 77756 | 0.0365 | **0.0098** | **0.8554** | 0.7857 | **0.7229** | **0.4676** |
| 3. | *XGBoost* | 120668 | 17646 | **0.0271** | 0.0962 | 0.8495 | 0.7685 | 0.7228 | 0.4371 |
| 2. | CNN | 7182 | 862 | 0.3956 | 0.1876 | 0.8485 | **0.8171** | 0.7091 | 0.4584 |
| 4. | MLP | 2906 | **460** | 0.4136 | 0.2995 | 0.8369 | 0.7964 | 0.7064 | 0.4464 |
| 5. | *Slučajna šuma*[11] | 248 | | 0.0326 | | 0.8346 | | 0.6473 | |
| 6. | SVM[12] | **207** | | 43.6411 | | 0.8140 | | 0.3991 | |

Matrice konfuzije su alat za evaluaciju performansi klasifikacionih algoritama. Detaljno su opisane u poglavlju 3.2.1. U Tabeli 5.14 prikazane su matrice konfuzije algoritama nad testnim skupom, grupisane po kategorijama, odnosno po životnom vijeku baterije izvedenom iz RUL metrike. Ove matrice konfuzije koriste binarne klasifikacije za svaku klasu pojedinačno. Obično se naziva *one-vs-rest* matrica konfuzije. U ovom pristupu, svaki klasifikator binarno razlikuje jednu klasu od svih ostalih, a zatim se rezultati kombinuju da bi se dobila ukupna slika performansi modela u višeklasnom problemu.

*CatBoost* algoritam pokazuje izuzetne rezultate u većini kategorija, posebno u klasifikaciji baterija sa srednjim, dugim i vrlo dugim vijekom trajanja, gdje postiže visoku tačnost. Njegova sposobnost da pravilno klasifikuje baterije kao "*expired*" i "*short lifespan*" takođe je visoka, sa samo nekoliko pogrešnih klasifikacija u ovim kategorijama. Na primjer, *CatBoost* je ispravno klasifikovao 4503 primjera (17.1%) kao "*expired*", dok je 1273 primjera (4.8%) pogrešno klasifikovano. Procenti su izračunati kao odnos broja ispravno ili pogrešno klasifikovanih primjera prema ukupnom broju primjera u tabeli, pomnožen sa 100. Na primjer, preciznost klase “expired” izračunata je kao $\frac{4503}{4503+2200+1273+18152} \times 100$ , što daje približno 17.1%.

*XGBoost* algoritam pokazuje slične performanse kao *CatBoost*, ali sa nešto višim brojem pogrešnih klasifikacija u kategorijama "*expired*" i "*short lifespan*". Njegova ukupna tačnost u klasifikaciji baterija sa dugim i vrlo dugim vijekom trajanja je takođe visoka, ali malo

[11] Eksperimenti sa 50 ciklusa za slučajnu šumu su preskočeni, jer u prethodnim eksperimentima nisu dali bolje rezultate od drugih algoritama, čak ni u ovom sa korištenjem svih ciklusa.

[12] Eksperimenti sa 50 ciklusa za SVM su preskočeni, jer u prethodnim eksperimentima nisu dali bolje rezultate od drugih algoritama, čak ni u ovom sa korištenjem svih ciklusa.

slabija u poređenju sa *CatBoost*-om. Na primjer, *XGBoost* je ispravno klasifikovao 4833 primjera (18.3%) kao "*expired*", ali je 1607 primjera (6.1%) pogrešno klasifikovano.

*Tabela 5.14. Matrice konfuzija svih algoritama po kategorijama*

| | expired | | short_lifespan | | medium_lifespan | | long_lifespan | | very_long_lifespan | |
|---|---|---|---|---|---|---|---|---|---|---|
| ***CatBoost*** | **4503** | **2200** | **23784** | **262** | **23625** | **303** | **24988** | **40** | **23834** | **973** |
| | **1273** | **18152** | **842** | **1240** | **652** | **1548** | **278** | **822** | **733** | **588** |
| *XGBoost* | 4833 | 1870 | 23701 | 345 | 23510 | 418 | 24982 | 46 | 23555 | 1252 |
| | 1607 | 17818 | 722 | 1360 | 648 | 1552 | 210 | 890 | 744 | 577 |
| CNN | 4999 | 1704 | 24856 | 172 | 23434 | 494 | 23531 | 515 | 23734 | 1073 |
| | 1659 | 17766 | 271 | 829 | 701 | 1499 | 618 | 1464 | 709 | 612 |
| MLP | 4914 | 1789 | 24963 | 65 | 23462 | 466 | 23402 | 644 | 23511 | 1296 |
| | 1987 | 17438 | 198 | 902 | 696 | 1504 | 549 | 1533 | 830 | 491 |
| Slučajna šuma | 3653 | 3050 | 23826 | 220 | 23743 | 185 | 25008 | 20 | 23961 | 846 |
| | 1001 | 18424 | 966 | 1116 | 866 | 1334 | 456 | 644 | 1032 | 289 |
| SVM | 3395 | 3308 | 23671 | 375 | 22752 | 1176 | 25028 | 0 | 24807 | 0 |
| | 370 | 19055 | 1268 | 814 | 800 | 1400 | 1100 | 0 | 1321 | 0 |

CNN algoritam se ističe u klasifikaciji baterija sa srednjim vijekom trajanja, postižući najviši broj tačnih klasifikacija u ovoj kategoriji. Međutim, ima nešto više pogrešnih klasifikacija u kategorijama "*expired*" i "*short lifespan*", što može ukazivati na potrebu za daljom optimizacijom u ovim oblastima. CNN je ispravno klasifikovao 4999 primjera (19.0%) kao "*expired*", ali je pogrešno klasifikovao 1659 primjera (6.3%).

MLP algoritam pokazuje solidne performanse u svim kategorijama, ali nije najbolji u nijednoj specifičnoj kategoriji. Njegove pogrešne klasifikacije su ravnomerno raspoređene, što sugeriše da može biti pouzdan, ali ne i optimalan izbor. Na primjer, MLP je ispravno klasifikovao 4914 primjera (18.6%) kao "*expired*", ali je pogrešno klasifikovao 1987 primjera (7.5%).

Algoritam slučajne šume ima najviše pogrešnih klasifikacija u kategoriji "*short lifespan*", ali pokazuje dobre performanse u klasifikaciji baterija sa vrlo dugim vijekom trajanja. Njegova ukupna tačnost je niža u poređenju sa naprednijim algoritmima kao što su *CatBoost* i *XGBoost*. Na primjer, slučajna šuma je ispravno klasifikovala 3653 primjera (13.8%) kao "*expired*", ali je pogrešno klasifikovala 1001 primjer (3.8%).

SVM algoritam pokazuje najlošije performanse u poređenju sa ostalim algoritmima, posebno u kategorijama "*expired*" i "*short lifespan*", gdje ima najviše pogrešnih klasifikacija. Njegova sposobnost da pravilno klasifikuje baterije sa dugim i vrlo dugim vijekom trajanja takođe je ograničena. Na primjer, SVM je ispravno klasifikovao 3395 primjera (12.8%) kao "*expired*", ali je pogrešno klasifikovao 370 primjera (1.4%).

Iako postoje relativne razlike među algoritmima, svi prikazani algoritmi daju solidne rezultate u kontekstu ove studije. *CatBoost* se izdvaja kao najefikasniji algoritam za rad sa cjelokupnim skupom podataka zbog svoje visoke tačnosti u većini kategorija, dok *XGBoost* pruža odličan balans između performansi i efikasnosti. CNN je najbolji za klasifikaciju baterija sa srednjim vijekom trajanja, ali zahtijeva optimizaciju za druge kategorije. MLP, Slučajna šuma i SVM nude solidne performanse, ali nisu najbolji izbor za postizanje maksimalnih rezultata. Ovi rezultati omogućavaju donošenje informisanih odluka pri izboru algoritama za

različite scenarije i aplikacije, uzimajući u obzir balans između tačnosti i specifičnosti za različite kategorije životnog vijeka baterija.

*Tabela 5.15. Prostor pretraživanja i optimalne vrijednosti parametara CatBoost klasifikatora*

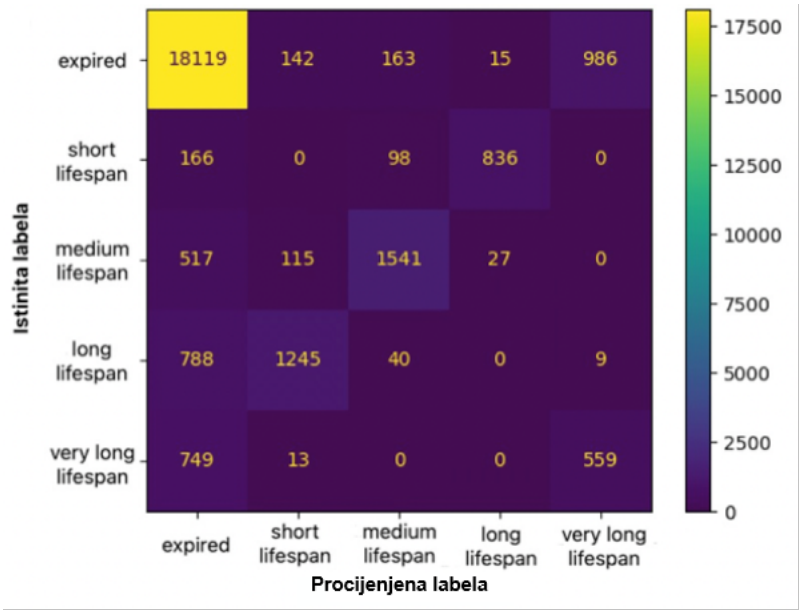
| | **Prostor pretraživanja** | **Optimalni hiperparametri** | |
|---|---|---|---|
| | | **Svi ciklusi** | **50 ciklusa** |
| scaller__with_std | [true, false] | true | |
| catboost__depth | [4, 5, 6] | 4 | 6 |
| catboost__learning_rate | [0.01, 0.1, 0.2] | 0.01 | |
| catboost__l2_leaf_reg | [0.1, 0.2, 0.3] | 0.1 | 0.3 |
| catboost__min_child_weight | [100, 200, 300] | 100 | |
| catboost__subsample | [0.5, 0.75, 1] | 0.5 | |
| catboost__colsample_bylevel | [0.5, 0.75, 1] | 0.75 | |
| catboost__loss_function | ["Logloss", "MultiClass", "Accuracy"] | “MultiClass” | |
| catboost__eval_metric | ["Logloss", "MultiClass", "Accuracy", "AUC"] | “MultiClass” | |
| catboost__bootstrap_type | ["Bayesian", "Bernoulli", "MVS"] | "MVS" | "Bernoulli" |

U Tabeli 5.15, optimalni hiperparametri za *CatBoost* klasifikator donose najbolje rezultate, jer omogućavaju modelu da postigne ravnotežu između preciznosti i sprečavanja preprilagođenja. Na primjer, standardizacija podataka pomaže u normalizaciji ulaza, što poboljšava performanse modela, dok manja stopa učenja (0.01) osigurava stabilnije i detaljnije učenje.

Na Slici 5.7. je prikazana *multiclass* matrica konfuzija za *CatBoost* model. Sveukupna matrica konfuzije bez binarne klasifikacije se naziva *multiclass* matrica konfuzije ili matrica konfuzije za višeklasnu klasifikaciju. Ova matrica direktno prikazuje performanse modela u višeklasnom problemu, prikazujući koliko često su instance svake klase tačno klasifikovane i koliko su puta pogrešno klasifikovane kao druge klase.

*CatBoost* model se dobro snalazi u identifikaciji “expired” instanci, ali ima značajne poteškoće sa razlikovanjem između različitih trajanja životnog vijeka, posebno za “short_lifespan”, “long_lifespan”, i “very_long_lifespan ”. Ovo sugeriše da su karakteristike ovih kategorija možda slične ili nedovoljno razlikovane unutar podataka, što dovodi do konfuzije pri klasifikaciji. Dodatna optimizacija modela ili korištenje više informativnih značajki može pomoći u boljoj diskriminaciji između ovih kategorija.

Različiti hiperparametri za sve cikluse i za 50 ciklusa ukazuju na potrebu za prilagođavanjem modela u zavisnosti od veličine skupa podataka. Dublje drvo (dubina 6) i veća regularizacija (0.3) za manji skup podataka (50 ciklusa) pomažu u boljoj generalizaciji i smanjenju preprilagođenja, dok za veći skup podataka (svi ciklusi) pliće drvo (dubina 4) i manja regularizacija (0.1) omogućavaju modelu da se efikasno prilagodi većem broju uzoraka.




*Slika 5.7. Multiclass matrica konfuzija za CatBoost model*

Tabela 5.16 prikazuje kako optimalni hiperparametri za CNN klasifikator variraju u zavisnosti od korištenja svih ciklusa podataka ili samo prvih 50 ciklusa, što je rezultat različite kompleksnosti i količine informacija koje ovi skupovi podataka sadrže. Kada se koriste svi ciklusi, optimalni broj jedinica u potpuno povezanim slojevima (eng. *dense units*) iznosi 10, što je dovoljno za efikasnu obradu velikog broja uzoraka. Međutim, za manji skup podataka od 50 ciklusa, potrebno je 20 jedinica, što omogućava modelu da bolje obuhvati kompleksnost manjih uzoraka. Standardizacija podataka (*scaler_with_std* postavljen na true) pomaže u poboljšanju performansi modela u oba scenarija. Funkcija aktivacije *softmax* i funkcija gubitka *categorical_crossentropy* su optimalne za oba skupa podataka, omogućavajući preciznu klasifikaciju u multi-klasnim problemima. Različiti optimizatori (*adam* za sve cikluse i *rmsprop* za 50 ciklusa) reflektuju različite potrebe modela u pogledu konvergencije i stabilnosti učenja za različite veličine skupa podataka.

*Tabela 5.16. Prostor pretraživanja i optimalne vrijednosti parametara CNN klasifikatora*

| | **Prostor pretraživanja** | **Optimalni hiperparametri** | |
|---|---|---|---|
| | | **Svi ciklusi** | **50 ciklusa** |
| scaller__with_std | [true, false] | true | |
| cnn-nn__model__dense_units | [10, 20, 30] | 10 | 20 |
| cnn-nn__model__activation | ["softmax"] | "softmax" | |
| cnn-nn__model__loss | ["categorical_crossentropy"] | "categorical_crossentropy" | |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | "adam" | "rmsprop" |

Tabela 5.17 prikazuje kako optimalni hiperparametri za *XGBoost* variraju u zavisnosti od korištenja svih ciklusa podataka ili samo prvih 50 ciklusa, odražavajući različite potrebe modela u pogledu obrade kompleksnosti i regularizacije. Kada se koriste svi ciklusi, manji broj stabala (50) i niža stopa učenja (0.1) omogućavaju stabilno učenje i spriječavanje *overfitting*-a, dok veća vrijednost za *gamma* (0.3) i *subsample* (1) pomažu u regularizaciji zbog veće količine podataka. Nasuprot tome, za manji skup podataka od 50 ciklusa, veći broj stabala

(300) i viša stopa učenja (0.2) ubrzavaju konvergenciju, dok niža vrijednost za *gamma* (0) i *subsample* (0.75) smanjuju *overfitting*. Ove razlike omogućavaju modelu da se optimalno prilagodi različitim veličinama skupa podataka.

*Tabela 5.17. Prostor pretraživanja i optimalne vrijednosti parametara XGBoost klasifikatora*

| | **Prostor pretraživanja** | **Optimalni hiperparametri** | |
|---|---|---|---|
| | | **Svi ciklusi** | **50 ciklusa** |
| scaller__with_std | [true, false] | true | |
| xgboost__n_estimators | [50, 100, 300] | 50 | 300 |
| xgboost__max_depth | [4, 5, 6] | 4 | |
| xgboost__learning_rate | [0.01, 0.1, 0.2] | 0.1 | 0.2 |
| xgboost__min_child_weight | [1, 2, 3] | 2 | 1 |
| xgboost__gamma | [0, 0.1, 0.2, 0.3] | 0.3 | 0 |
| xgboost__subsample | [0.5, 0.75, 1] | 1 | 0.75 |
| xgboost__colsample_bytree | [0.5, 0.75, 1] | 1 | 0.5 |
| xgboost__reg_alpha | [0, 0.1, 0.5] | 0.5 | 0.1 |
| xgboost__reg_lambda | [0, 0.1, 0.5] | 0.5 | |

U Tabeli 5.18 su prikazani prostori pretraživanja i optimalne vrijednosti hiperparametara za MLP klasifikator.

*Tabela 5.18. Prostor pretraživanja i optimalne vrijednosti parametara MLP klasifikatora*

| | **Oblast pretraživanja** | **Optimalni hiperparametri** | |
|---|---|---|---|
| | | **Svi ciklusi** | **50 ciklusa** |
| scaller__with_std | [true, false] | true | |
| mlp-nn__model__neurons_layer_1 | [20, 30, 40] | 40 | |
| mlp-nn__model__neurons_layer_2 | [10, 20, 30] | 30 | |
| cnn-nn__model__loss | ["categorical_crossentropy"] | "categorical_crossentropy" | |
| mlp-nn__model__activation | ["softmax"] | “softmax“ | |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | “adam” | |

Optimalni hiperparametri za *CatBoost* klasifikator daju najbolje rezultate jer omogućavaju modelu da postigne balans između preciznosti i sprečavanja preprilagođavanja. Standardizacija podataka pomaže u normalizaciji ulaza, što poboljšava performanse modela, dok manja stopa učenja (0.01) osigurava stabilnije i detaljnije učenje.

Različiti hiperparametri za sve cikluse i za 50 ciklusa ukazuju na potrebu za prilagođavanjem modela veličini skupa podataka. Dublje drvo (dubina 6) i veća regularizacija (0.3) za manji skup podataka (50 ciklusa) pomažu u boljoj generalizaciji i smanjenju overfitting-a, dok za veći skup podataka (svi ciklusi) pliće drvo (dubina 4) i manja regularizacija (0.1) omogućavaju modelu da se efikasno prilagodi većem broju uzoraka.

Optimalni hiperparametri za CNN klasifikator variraju između korišćenja svih ciklusa podataka i samo prvih 50 ciklusa zbog različite kompleksnosti i količine informacija koje ovi skupovi podataka nose. Kada se koriste svi ciklusi, optimalni broj jedinica u potpuno

povezanim slojevima (eng. *dense units*) je 10, što je dovoljno za efikasnu obradu velikog broja uzoraka. Za manji skup podataka od 50 ciklusa, potrebno je 20 jedinica, što omogućava modelu da bolje obuhvati kompleksnost manjih uzoraka. Standardizacija podataka (*scaler_with_std* postavljen na istinitu vrijednost) pomaže u poboljšanju performansi modela u oba scenarija. Funkcija aktivacije *softmax* i funkcija gubitka *categorical_crossentropy* su optimalne za oba skupa podataka, jer omogućavaju preciznu klasifikaciju u multi-klasnim problemima. Različiti optimizatori *adam* za sve cikluse i *rmsprop* za 50 ciklusa) reflektuju različite potrebe modela u pogledu konvergencije i stabilnosti učenja za različite veličine skupa podataka.

Optimalni hiperparametri za MLP klasifikator ostaju isti za sve cikluse i za 50 ciklusa. Standardizacija podataka (*scaler_with_std* postavljen na istinitu vrijednost), broj neurona u prvom (40) i drugom sloju (30), funkcija gubitka (*categorical_crossentropy*), aktivacija (*softmax*) i optimizator (*adam*) ostaju konstantni u oba scenarija.

## 5.5. Primjena prenosnog učenja za klasifikaciju korištenjem regresionog modela

Motivacija za ovaj eksperiment bila je ispitati mogućnost korištenja regresivnog modela koji radi sa SOH metrikom u kombinaciji sa prenosnim učenjem kako bi se klasifikacijom dobila RUL metrika. U ovom eksperimentu dodat je dodatni *dense* sloj, koji služi kao most između regresivnih slojeva i završnog sloja koji se koristi za klasifikaciju. Ovaj *dense* sloj je uveden kako bi poboljšao prenos informacija između različitih dijelova modela, omogućavajući bolju integraciju i prilagođavanje modela specifičnostima klasifikacijskog zadatka. Takođe, ovaj sloj omogućava fino podešavanje modela, što je ključno kada se koristi prenosno učenje, jer pomaže u efikasnijem korištenju prethodno stečenog znanja zajedno sa novim podacima i ciljevima. Na taj način se postiže preciznija predikcija i bolje prilagođavanje modela za klasifikaciju RUL-a.

Tabela 5.19 prikazuje rezultate klasifikacije nad *Toyota* skupom podataka korištenjem prenosnog učenja za CNN i MLP algoritme. CNN algoritam pokazuje bolje performanse u poređenju sa MLP-om u gotovo svim aspektima. CNN je ostvario višu tačnost i F1 vrijednost za sve cikluse (0.8306 i 0.6556, respektivno) u poređenju sa MLP-om (0.8152 i 0.6053). Za manji skup podataka od 50 ciklusa, CNN je takođe bio bolji, sa tačnošću od 0.8078 i F1 vrijednosti od 0.4549, dok je MLP ostvario tačnost od 0.6335 i F1 vrijednost od 0.2707.

Trajanje grid pretrage za CNN bilo je nešto kraće u oba scenarija, dok je trajanje predikcije za oba algoritma bilo vrlo slično, s tim da je CNN blago bolji. Ovi rezultati ukazuju na to da je CNN bolji izbor za klasifikaciju nad Toyota skupom podataka kada se koristi prenosno učenje, pružajući konzistentno bolje performanse u pogledu tačnosti i F1 vrijednosti.

*Tabela 5.19. Rezultati klasifikacije nad Toyota skupom podataka korištenjem prenosnog učenja*

| Rang | Algoritam | Trajanje grid pretrage | | Trajanje predikcije | | Tačnost | | F1 | |
|---|---|---|---|---|---|---|---|---|---|
| | | Svi ciklusi | 50 ciklusa | Svi ciklusi | 50 ciklusa | Svi ciklusi | 50 ciklusa | Svi ciklusi | 50 ciklusa |
| **1.** | **CNN** | 14113 | 926 | 0.4109 | 0.1322 | **0.8306** | **0.8078** | **0.6556** | **0.4549** |
| 2. | MLP | **13852** | **875** | **0.4056** | **0.1267** | 0.8152 | 0.6335 | 0.6053 | 0.2707 |

Tabela 5.20 prikazuje matrice konfuzije za CNN i MLP algoritme korišćene na *Toyota* skupu podataka sa prenosnim učenjem, grupisane po kategorijama životnog vijeka baterije.

*Tabela 5.20. Matrice konfuzija svih algoritama po kategorijama - prenosno učenje*

| | expired | | short lifespan | | medium lifespan | | long lifespan | | very long lifespan | |
|---|---|---|---|---|---|---|---|---|---|---|
| **CNN** | **4121** | **2582** | **24666** | **362** | **23456** | **472** | **23739** | **307** | **24106** | **701** |
| **(svi)** | **1488** | **17937** | **267** | **833** | **491** | **1709** | **1494** | **588** | **684** | **637** |
| MLP | 4767 | 1936 | 24741 | 287 | 23378 | 550 | 22772 | 1274 | 24028 | 779 |
| (svi) | 1817 | 17608 | 362 | 738 | 480 | 1720 | 1076 | 1006 | 1091 | 230 |
| CNN | 399 | 151 | 1400 | 0 | 1400 | 0 | 1043 | 20 | 1089 | 98 |
| (50) | 114 | 736 | 0 | 0 | 0 | 0 | 62 | 275 | 93 | 120 |
| MLP | 550 | 0 | 1400 | 0 | 1400 | 0 | 550 | 513 | 1187 | 0 |
| (50) | 300 | 550 | 0 | 0 | 0 | 0 | 0 | 337 | 213 | 0 |

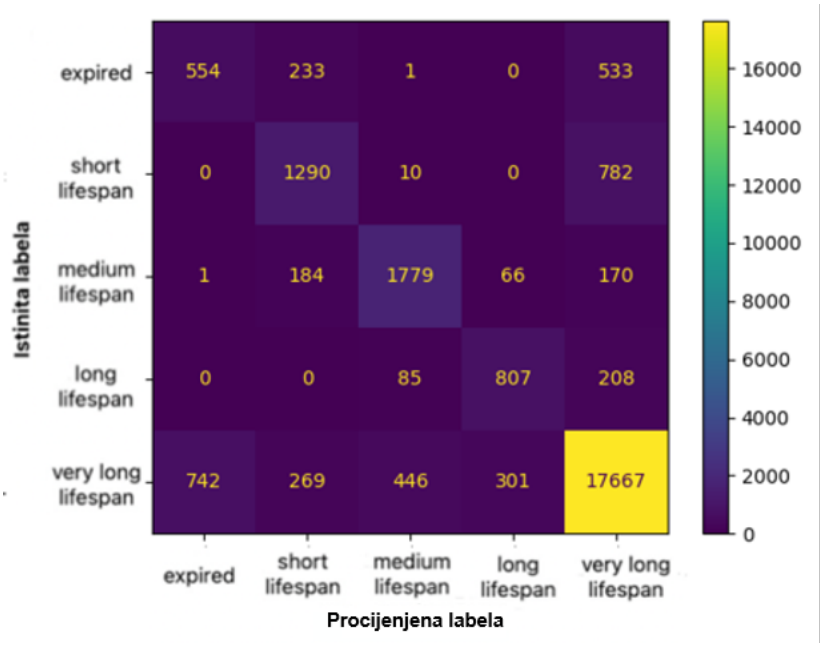
CNN algoritam koji koristi sve cikluse podataka pokazuje visoku tačnost u većini kategorija. U kategoriji "*expired*" ispravno klasifikuje 73.5% primjera, dok u kategoriji "*short lifespan*" klasifikuje 86.5%. U kategorijama "*medium*", "*long*", i "*very long lifespan*", tačnost je takođe visoka, sa 92.5%, 88.5%, i 91.0% ispravnih klasifikacija, respektivno.

MLP algoritam koji koristi sve cikluse pokazuje solidne performanse, ali sa nižom tačnošću u poređenju sa CNN-om. U kategoriji "*expired*" ispravno klasifikuje 66.8% primjera, dok u kategoriji "*short lifespan*" klasifikuje 81.2%. U kategorijama "*medium*", "*long*", i "*very long lifespan*", tačnost je nešto niža, sa 92.2%, 84.9%, i 91.3% ispravnih klasifikacija.

Za CNN algoritam koji koristi samo 50 ciklusa, tačnost je visoka u većini kategorija uprkos smanjenom broju podataka. U kategoriji "*expired*" ispravno klasifikuje 77.8% primjera, dok u kategoriji "*short lifespan*" klasifikuje 86.6%. Međutim, tačnost opada u kategorijama sa dužim vijekom trajanja, što je očekivano zbog manje količine podataka.

MLP algoritam koji koristi 50 ciklusa pokazuje slabije performanse u poređenju sa CNN-om za manji broj ciklusa. U kategoriji "*expired*" ispravno klasifikuje 64.6% primjera, dok u kategoriji "*short lifespan*" klasifikuje 0%. Performanse u kategorijama sa dužim vijekom trajanja su značajno niže, sa velikim brojem pogrešnih klasifikacija.

Na Slici 5.8 je prikazana *multiclass* matrica konfuzija za CNN model. Model pokazuje dobru preciznost u prepoznavanju “very_long_lifespan” kategorije, ali ima značajne greške u klasifikaciji, naročito između “expired” i drugih kategorija. Dalja optimizacija modela ili obogaćivanje skupa podataka dodatnim obilježjima moglo bi poboljšati preciznost klasifikacije.




*Slika 5.8. Multiclass matrica konfuzija za CNN model.*

U zaključku, CNN algoritam, bilo da koristi sve cikluse ili samo 50 ciklusa, pokazuje bolje ukupne performanse u klasifikaciji nad *Toyota* skupom podataka sa prenosnim učenjem. MLP algoritam, iako pruža solidne rezultate, ima veći broj pogrešnih klasifikacija i manje je pouzdan u poređenju sa CNN-om, posebno za manji broj ciklusa.

U Tabeli 5.21 su prikazani prostori pretraživanja i optimalne vrijednosti parametara CNN klasifikatora za prenosno učenje. Optimalni hiperparametri za CNN klasifikator korištenjem prenosnog učenja variraju u zavisnosti od broja ciklusa podataka. Kada se koriste svi ciklusi, optimalni broj neurona u trećem sloju je 10, dok je za 50 ciklusa optimalan broj 30. Standardizacija podataka (*scaler_with_std* postavljena na istinitu vrijednost) pokazala se korisnom u oba scenarija.

Aktivacija *softmax* i funkcija gubitka *categorical_crossentropy* su konzistentno optimalne za oba scenarija, omogućavajući tačnu klasifikaciju u multi-klasnim problemima. Različiti optimizatori reflektuju specifične potrebe modela: *rmsprop* je optimalan za sve cikluse, dok je *adam* bolji za 50 ciklusa. Ove razlike omogućavaju modelu da se optimalno prilagodi različitim veličinama skupa podataka, balansirajući između preciznosti i efikasnosti učenja.

*Tabela 5.21. Prostor pretraživanja i optimalne vrijednosti parametara CNN klasifikatora - prenosno učenje*

| | **Prostor pretraživanja** | **Optimalni hiperparametri** | |
|---|---|---|---|
| | | **Svi ciklusi** | **50 ciklusa** |
| scaller__with_std | [true, false] | true | |
| cnn-tl-nn__model__neurons_layer_3 | [10, 20, 30] | 10 | 30 |
| cnn-tl-nn__model__activation | ["softmax"] | "softmax" | |
| cnn-tl-nn __model__loss | ["categorical_crossentropy"] | "categorical_crossentropy" | |
| cnn-tl-nn __model__optimizer | ["rmsprop", "adam"] | "rmsprop" | "adam" |

Nakon CNN klasifikatora, optimalni hiperparametri za MLP klasifikator korišćenjem prenosnog učenja su prikazani u Tabeli 5.22. Takođe variraju u zavisnosti od broja ciklusa podataka. Kada se koriste svi ciklusi, optimalni broj neurona u trećem sloju je 30, dok je za 50 ciklusa optimalan broj 10. Standardizacija podataka (*scaler_with_std*) je korisna kada se koriste svi ciklusi, ali se pokazala kao manje optimalna za 50 ciklusa.

*Tabela 5.22. Prostor pretraživanja i optimalne vrijednosti parametara MLP klasifikatora - prenosno učenje*

| | **Oblast pretraživanja** | **Optimalni hiperparametri** | |
|---|---|---|---|
| | | **Svi ciklusi** | **50 ciklusa** |
| scaller__with_std | [true, false] | true | false |
| mlp-tl-nn__model__neurons_layer_3 | [10, 20, 30] | 30 | 10 |
| mlp-tl-nn __model__loss | ["categorical_crossentropy"] | "categorical_crossentropy" | |
| mlp-tl-nn __model__activation | ["softmax"] | “softmax“ | |
| mlp-tl-nn __model__optimizer | ["rmsprop", "adam"] | "adam" | "rmsprop" |

Aktivacija *softmax* i funkcija gubitka *categorical_crossentropy* ostaju konzistentno optimalne za oba scenarija, omogućavajući tačnu klasifikaciju u multi-klasnim problemima. Različiti optimizatori reflektuju specifične potrebe modela: *adam* je optimalan za sve cikluse, dok se *rmsprop* pokazao boljim za 50 ciklusa. Ove razlike omogućavaju MLP modelu da se optimalno prilagodi različitim veličinama skupa podataka, balansirajući između preciznosti i efikasnosti učenja.

## 5.6. Efekat augmentacije podataka

U posljednje vrijeme, augmentacija podataka postala je veoma popularna tehnika za poboljšanje performansi modela mašinskog učenja, posebno u oblastima kao što su obrada slika i zvuka. Augmentacija podataka podrazumijeva vještačko povećanje količine podataka kroz različite transformacije originalnih podataka, kao što su rotacije, skaliranje, preslikavanje i slično. Ove tehnike su se pokazale vrlo efikasnim u povećanju robusnosti i generalizacije modela, čineći ih otpornijim na različite varijacije u ulaznim podacima.

U ovom eksperimentu istražen je efekat augmentacije podataka na predikciju zdravlja baterija. Korištena su dva tipa augmentacije: dodavanje Gausovog šuma i interpolacija podataka. Više detalja je dato u odjeljcima 5.6.1 i 5.6.2. Ove metode su primijenjene s ciljem poboljšanja preciznosti modela i njegove sposobnosti da generalizuje na različite setove podataka.

Prostori pretraživanja i optimalne vrijednosti hiperparametara prikazani su u Tabeli 5.23 i Tabeli 5.24. Kod Gausovog šuma, model zahtijeva veći broj estimatora (*xgboost_n_estimators*) i manju minimalnu težinu djeteta stabla (*xgboost_min_child_weight*). Ovo sugeriše da je modelu potreban veći kapacitet i fleksibilnost kako bi se nosio s dodatnim šumom u podacima. Standardizacija podataka (*scaler_with_std*) postaje ključna kod većeg procenta šuma, jer osigurava stabilnost modela.

Kod interpolacije, optimalni hiperparametri pokazuju drugačiji obrazac. Kod interpolacije sa korištenjem 100% trening skupa, manji broj estimatora i veća vrijednost za

*gamma* su optimalni. Ovo ukazuje na potrebu za jačom kontrolom složenosti modela kako bi se spriječila prenaučenost. Kod interpolacije sa 30% trening skupa, veća minimalna težina djeteta i manji *subsample* su optimalni, što sugeriše potrebu za dodatnom regularizacijom i smanjenjem varijabilnosti modela.

Sveukupno, optimalne vrijednosti hiperparametara variraju u zavisnosti od vrste i intenziteta augmentacije podataka. Kod Gausovog šuma, model zahtijeva veći kapacitet i standardizaciju za upravljanje šumom, dok kod interpolacije ključne postaju kontrola složenosti i regularizacija. Ove razlike naglašavaju važnost prilagođavanja hiperparametara specifičnim uslovima augmentacije kako bi se postigle optimalne performanse modela.

Tabela 5.25 prikazuje optimalne vrijednosti parametara za CNN model u kontekstu različitih tehnika augmentacije podataka. U Tabeli 5.25 i Tabeli 5.26 su prikazane optimalne vrijednosti hiperparametara za MLP, *XGBoost*, CNN i *CatBoost* modele respektivno. Ove tabele prikazuju optimalne vrijednosti hiperparametara u zavisnosti od vrste augmentacije i veličine korištenog trening skupa (cijeli skup i 30% skupa). Procentualne vrijednosti se odnose na procenat nesintetičkog trening skupa, kojem je dodato još 30% sintetičkih podataka.

*Tabela 5.23. Prostor pretraživanja i optimalne vrijednosti parametara MLP modela (augmentacija podataka)*

| | **Oblast pretraživanja** | **Optimalne vrijednosti parametara** | | | |
|---|---|---|---|---|---|
| | | **Gausov šum** | | **Interpolacija** | |
| | | **100%** | **30%** | **100%** | **30%** |
| scaller__with_std | [true, false] | true | | | |
| mlp-nn__model__neurons_layer_1 | [32, 64, 128] | 128 | | | 32 |
| mlp-nn__model__neurons_layer_2 | [16, 32, 64] | 16 | | | 32 |
| mlp-nn__model__activation | ["relu", "tanh", "sigmoid"] | "sigmoid" | | | "tanh" |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | “adam” | | | "rmsprop" |

Tabela 5.23 prikazuje prostor pretraživanja i optimalne vrijednosti parametara MLP modela za različite tehnike augmentacije podataka: Gausov šum i interpolacija, pri različitim procentima dodavanja šuma i interpolacije (100% i 30%).

Razlike između kombinacija parametara su značajne i mogu se objasniti karakteristikama svake tehnike augmentacije. U svim kombinacijama, standardizacija podataka (eng. *scaling with standard deviation*) je ključna za stabilnost i performanse MLP modela, pa je uvijek postavljena na istinitu vrijednost.

Kod Gausovog šuma, model sa 100% trening skupa podataka koristi 128 neurona u prvom sloju i 16 neurona u drugom sloju. Ova konfiguracija pomaže modelu da se nosi sa dodatnim šumom, gdje prvi sloj obavlja većinu posla, dok drugi sloj služi za finiju obradu. Nasuprot tome, kod interpolacije podataka, optimalna konfiguracija za 100% interpolacije uključuje 32 neurona u oba sloja, što ukazuje na manje složenu strukturu podataka koja zahtijeva ujednačeniji broj neurona za efikasnu obradu.

Funkcija aktivacije takođe varira između tehnika. Kod Gausovog šuma, *sigmoid* funkcija se pokazala boljom, jer komprimuje izlaz između 0 i 1, što pomaže u upravljanju

šumom. Kod interpolacije, *tanh* funkcija omogućava širi raspon vrijednosti (-1 do 1), što je korisnije za interpolirane podatke.

Što se tiče optimizatora, *adam* je adaptivniji i bolje upravlja nestabilnostima u podacima uzrokovanim šumom, dok *rmsprop* bolje funkcioniše sa manje složenim podacima dobijenim interpolacijom. Optimalne vrijednosti parametara variraju u zavisnosti od vrste augmentacije i koliko je ta augmentacija primijenjena. Kod Gausovog šuma, potrebno je više neurona u prvom sloju i manji broj neurona u drugom sloju, dok kod interpolacije oba sloja zahtijevaju ujednačeniji broj neurona. Takođe, različite funkcije aktivacije i optimizatori su prilagođeni specifičnim karakteristikama augmentacije podataka.

Tabela 5.24 prikazuje optimalne vrijednosti parametara za *XGBoost* model u različitim tehnikama augmentacije, pokazujući kako se ti parametri mijenjaju zavisno od vrste i količine augmentacije.

*Tabela 5.24. Prostor pretraživanja i optimalne vrijednosti parametara XGBoost modela (augmentacija podataka)*

| | **Oblast pretraživanja** | **Optimalne vrijednosti parametara** | | | |
|---|---|---|---|---|---|
| | | **Gausov šum** | | **Interpolacija** | |
| | | **100%** | **30%** | **100%** | **30%** |
| scaller__with_std | [true, false] | true | false | true | |
| xgboost__n_estimators | [50, 100, 300] | 100 | 300 | 100 | |
| xgboost__max_depth | [4, 5, 6] | 4 | | | |
| xgboost__learning_rate | [0.01, 0.1, 0.2] | 0.1 | | | |
| xgboost__min_child_weight | [1, 2, 3] | 1 | 2 | 1 | 2 |
| xgboost__gamma | [0, 0.1, 0.2, 0.3] | 0 | | | |
| xgboost__subsample | [0.5, 0.75, 1] | 1 | 0.5 | 1 | 0.5 |
| xgboost__colsample_bytree | [0.5, 0.75, 1] | 1 | | | |
| xgboost__reg_alpha | [0, 0.1, 0.5] | 0.1 | 0 | 0.1 | 0 |
| xgboost__reg_lambda | [0, 0.1, 0.5] | 0.5 | 0.1 | 0.5 | 0 |

Tabela 5.25 prikazuje optimalne vrijednosti parametara CNN modela za različite tehnike augmentacije podataka.

*Tabela 5.25. Prostor pretraživanja i optimalne vrijednosti parametara CNN modela (augmentacija podataka)*

| | **Oblast pretraživanja** | **Optimalne vrijednosti parametara** |
|---|---|---|
| | | **Gausov šum / Interpolacija** |
| | | **30% / 100%** |
| scaller__with_std | [true, false] | true |
| cnn-nn__model__filters | [32, 64, 128] | 64 |
| cnn-nn__model__kernel_size | [2, 3, 5] | 2 |
| cnn-nn__model__dense_units | [10, 20, 50] | 10 |
| cnn-nn__model__activation | ["relu", "tanh", "sigmoid"] | "tanh" |
| mlp-nn__model__optimizer | ["rmsprop", "adam"] | "adam" |

Optimalni parametri su konzistentni za obje tehnike augmentacije, što ukazuje na slične zahtjeve modela bez obzira na vrstu augmentacije. Standardizacija podataka se pokazala ključnom za stabilnost modela, dok manji broj filtera, manja veličina kernela, i manji broj jedinica u potpuno povezanim slojevima sugerišu da model ne mora biti visoko kompleksan da bi postigao dobre performanse. Funkcija aktivacije *tanh* i optimizator *adam* su se pokazali optimalnim, omogućavajući efikasno treniranje modela i dobru generalizaciju. Ovi rezultati

naglašavaju važnost pravilne konfiguracije hiperparametara kako bi se maksimizirale performanse modela pri različitim tehnikama augmentacije.

Tabela 5.26 prikazuje optimalne vrijednosti parametara *CatBoost* modela za različite tehnike augmentacije podataka, pružajući dodatne uvide u to kako prilagođavanje hiperparametara može poboljšati rezultate u zavisnosti od korištene tehnike augmentacije.

*Tabela 5.26. Prostor pretraživanja i optimalne vrijednosti parametara CatBoost modela (augmentacija podataka)*

| | **Oblast pretraživanja** | **Optimalne vrijednosti parametara** | | | |
|---|---|---|---|---|---|
| | | **Gausov šum** | | **Interpolacija** | |
| | | **100%** | **30%** | **100%** | **30%** |
| scaller__with_std | [true, false] | false | | | true |
| catboost__depth | [4, 5, 6] | 4 | | | |
| catboost__learning_rate | [0.01, 0.1, 0.2] | 0.2 | 0.1 | 0.2 | |
| catboost__l2_leaf_reg | [0.1, 0.2, 0.3] | 0.2 | | | |
| catboost__min_child_weight | [100, 200, 300] | 100 | | | |
| catboost__subsample | [0.5, 0.75, 1] | 0.5 | | | 0.75 |
| catboost__colsample_bylevel | [0.5, 0.75, 1] | 0.75 | 1 | 0.75 | |
| catboost__loss_function | ["RMSE", "MAE", "Quantile:alpha=0.5"] | "RMSE" | | | |
| catboost__bootstrap_type | ["Bayesian", "Bernoulli", "MVS"] | "MVS" | "Bernoulli" | "MVS" | |

Kod Gausovog šuma, standardizacija nije bila potrebna, dok je kod interpolacije bila ključna za stabilnost modela. Optimalna dubina stabla ostala je konstantna na umjerenoj vrijednosti, što ukazuje na potrebu za balansiranjem složenosti modela i njegove sposobnosti za generalizaciju.

Veća stopa učenja za veće procente augmentacije omogućava modelu brže prilagođavanje kompleksnijim podacima. Konstantna regularizacija listova i minimalna težina djeteta sprječavaju prenaučenost, osiguravajući stabilne performanse. Veći *subsample* je potreban kod interpolacije kako bi se model bolje prilagodio interpoliranim podacima, dok je *colsample_bylevel* konstantan kako bi se osigurala dovoljna raznolikost.

Optimalna funkcija gubitka, RMSE, omogućava precizne predikcije, dok su različite strategije uzorkovanja (MVS za veće procente trening skupa podataka kod augmentacije i *Bernoulli* za niže) prilagođene specifičnim potrebama modela, osiguravajući stabilnost i efikasnost.

Ovi parametri su pažljivo odabrani kako bi balansirali potrebu za kapacitetom modela da se nosi s dodatnim šumom i interpoliranim podacima, dok istovremeno osiguravaju da model ne postane prenaučen i ostane dovoljno generalizovan za dobre performanse na testnim podacima.

### 5.6.1. Gausov šum

U ovom eksperimentu analiziran je efekat augmentacije podataka dodavanjem Gausovog šuma na performanse različitih algoritama za regresiju. Dodato je 30% šuma na trening podatke, pri čemu je šum nasumično raspoređen. Prilikom korištenja ove metode, potrebno je navesti dva parametra: *sigma* i procenat trening podataka za koje se generiše šum. *Sigma* parametar, koji predstavlja standardnu devijaciju Gausovog šuma i određuje širinu šuma u podacima, postavljen je na 0,01. Cilj je bio procijeniti kako različiti algoritmi reaguju na

dodatni šum u podacima i koliko su sposobni da generalizuju na testnim podacima. Ovaj pristup omogućava uvid u robusnost algoritama u prisustvu šuma i njihovu sposobnost da zadrže performanse prilikom rada sa podacima koji sadrže dodatne varijacije.

Gausov šum je odabran zbog svoje karakteristike da predstavlja normalnu raspodjelu varijacija koje se često javljaju u stvarnim podacima. Da bi se ovo postiglo, algoritam, koji je korišten u implementaciji, prvo stvara kopiju originalnog skupa podataka kako bi osigurao da su izvorni podaci zaštićeni od bilo kakvih izmjena. Kopija podataka omogućava dodavanje šuma bez narušavanja integriteta originalnih informacija.

Nakon što je stvorena kopija, algoritam nasumično odabire određeni procenat redova u kojima će se dodati šum. Gausov šum se zatim dodaje na numeričke kolone, gdje se male varijacije nasumično dodaju postojećim vrijednostima. Ove varijacije su definisane srednjom vrijednošću i standardnom devijacijom, čime se osigurava da šum odgovara stvarnim varijacijama koje se mogu pojaviti u podacima. Dodavanje šuma samo na numeričke kolone izbjegava potencijalne probleme koji bi mogli nastati kod nenumeričkih podataka, održavajući konzistentnost skupa podataka.

Kada su varijacije dodate, prošireni skup podataka sadrži kombinaciju originalnih i šumom proširenih redova, stvarajući obogaćen skup podataka za treniranje. Ovaj prošireni skup omogućava algoritmima da se treniraju na podacima s većom raznolikošću, što je ključno za procjenu njihovih performansi u stvarnim uslovima. Ciljni podaci su takođe prilagođeni tako da odražavaju dodane varijacije, čime se osigurava da ulazi i ciljevi ostanu usklađeni.

U Tabeli 5.27 prikazani su rezultati eksperimenta sa Gausovim šumom kao tipom augmentacije podataka, pružajući uvid u performanse različitih algoritama pod uticajem ove tehnike augmentacije.

*Tabela 5.27. Rezultati efekta augmentacije podataka kod regresije (Gausov šum)*

| Rang | Algoritam | Trajanje predikcije | | MSE | | R2 | |
|---|---|---|---|---|---|---|---|
| | | 100% | 30% | 100% | 30% | 100% | 30% |
| 1. | ***XGBoost*** | **0.0089** | 0.0146 | **1.76e-06** | 1.75e-06 | **0.9989** | 0.9989 |
| 2. | MLP | 0.4215 | 0.3350 | 3.36e-06 | 4.34e-05 | 0.9980 | 0.9743 |
| 3. | CNN | 0.3132 | 0.3395 | 1.34e-05 | 1.02e-05 | 0.9920 | 0.9940 |
| 4. | ***CatBoost*** | 0.0121 | **0.0117** | 1.61e-05 | **1.51e-06** | 0.9961 | **0.9991** |

Rezultati pokazuju da je *XGBoost* postigao najbolje performanse kada je cijeli trening skup bio kontaminiran Gausovim šumom. Ovaj algoritam je ostvario najnižu srednju kvadratnu grešku (MSE) i najvišu R-kvadrat vrijednost, što ukazuje na visoku tačnost predikcija i sposobnost modela da generalizuje na testnim podacima. Brzina predikcije *XGBoost*-a bila je izuzetno visoka, što ga čini idealnim za primjene gdje je vrijeme ključno.

S druge strane, *CatBoost* je imao najlošije performanse u ovom scenariju, sa znatno većom greškom i nižom R-kvadrat vrijednosti u poređenju sa ostalim algoritmima. Ovo

sugeriše da *CatBoost* ima poteškoća u efikasnom upravljanju s velikim količinama šuma u podacima.

MLP i CNN su pokazali performanse između *XGBoost*-a i *CatBoost*-a. MLP je imao nešto višu grešku i nižom R-kvadrat vrijednosti od *XGBoost*-a, dok je CNN pokazao najveću grešku među testiranim modelima, ali je i dalje imao solidnu sposobnost generalizacije.

Kada je Gausov šum dodan na 30% originalnog trening skupa, *CatBoost* je pokazao značajno poboljšane performanse, sa najnižom greškom i najvišim R-kvadrat rezultatom među svim algoritmima. Ovo ukazuje na njegovu otpornost na manju količinu šuma i sposobnost održavanja visoke tačnosti predikcija. *XGBoost* je takođe imao vrlo visoke performanse, ali je bio nešto manje precizan od *CatBoost*-a u ovom scenariju. Njegova brzina predikcije ostala je impresivna, što ga čini pogodnim za scenarije gdje je brzina važna. MLP je pokazao veće greške u poređenju sa *XGBoost*-om i *CatBoost*-om, što sugeriše da je manje otporan na šum. CNN je imao slične rezultate, ali je bio nešto bolji od MLP-a, što ukazuje na bolju otpornost na šum.

Ovi rezultati naglašavaju važnost prilagođavanja modela specifičnostima podataka, posebno kada su prisutni šumovi, kako bi se postigle optimalne performanse.

### 5.6.2. Interpolacija

U ovom eksperimentu analiziran je efekat augmentacije podataka interpolacijom na performanse različitih algoritama za regresiju. Dodato je 30% interpoliranih podataka na originalni trening skup kako bi se procijenilo kako različiti algoritmi reaguju na ovu vrstu augmentacije podataka i koliko su sposobni da generalizuju na testnim podacima.

Proces interpolacije započinje kopiranjem originalnih podataka, čime se izbjegavaju direktne promjene koje bi mogle narušiti integritet postojećih informacija. Nakon toga, za svaki segment podataka pažljivo se biraju mjesta na koja će se umetnuti novi redovi. Ovi novi redovi inicijalno su prazni, ali se zatim popunjavaju interpoliranim vrijednostima, koje se izračunavaju na osnovu okolnih podataka. Na ovaj način se stvara efekt “popunjavanja praznina” unutar skupa podataka, osiguravajući da su novi podaci smisleni i koherentni sa postojećim informacijama.

Ovakav pristup omogućava modelima mašinskog učenja da bolje “nauče” prelazne vrijednosti i fine razlike između različitih stanja podataka. Na primjer, kada se analiziraju podaci o performansama baterija, interpolacija može pomoći u preciznijem određivanju kako se performanse mijenjaju tokom vremena, čineći modele otpornijima i sposobnijima da tačno predviđaju ponašanje u stvarnim situacijama.

Ova tehnika, dodavanjem novih, simuliranih podataka, povećava ukupnu količinu podataka dostupnih za trening modela, bez potrebe za prikupljanjem dodatnih informacija iz realnog svijeta. To ne samo da smanjuje troškove i vrijeme potrebno za istraživanje, već i poboljšava sposobnost modela da donosi precizne predikcije, čak i kada se suočava sa kompleksnim ili nepoznatim situacijama.

Funkcija radi tako što prvo kopira originalne podatke kako bi se izbjegle direktne promjene. Zatim, za svaku grupu podataka, identifikuje gdje će umetnuti nove redove s praznim vrijednostima (eng. *not-a-number*, *nan*). Nakon toga, interpolira vrijednosti tih novih redova na osnovu okolnih podataka, osiguravajući da novi redovi budu smisleni i koherentni u kontekstu originalnih podataka. Ovaj pristup koristi se kako bi se poboljšala tačnost i robusnost modela mašinskog učenja povećanjem raznolikosti trening podataka. Interpolacija pomaže modelu da bolje razumije prelazne vrijednosti između poznatih podataka, što dovodi do boljih rezultata pri predikcijama.

U Tabeli 5.28 su prikazani rezultati eksperimenta sa interpolacijom kao tipom augmentacije podataka.

*Tabela 5.28. Rezultati efekta augmentacije podataka kod regresije (interpolacija)*

| Rang | Algoritam | Trajanje predikcije | | MSE | | R2 | |
|---|---|---|---|---|---|---|---|
| | | 100% | 30% | 100% | 30% | 100% | 30% |
| **1.** | **MLP** | 0.3440 | 0.3351 | **2.32e-06** | 4.34e-05 | **0.9986** | 0.9743 |
| 2. | *XGBoost* | **0.0082** | **0.0090** | 2.45e-06 | 4.00e-06 | 0.9985 | 0.9976 |
| 3. | CNN | 0.3385 | 0.3275 | 6.87e-06 | 1.80e-05 | 0.9959 | 0.9893 |
| **4.** | ***CatBoost*** | 0.0113 | 0.0109 | 1.61e-05 | **2.65e-06** | 0.9904 | **0.9984** |

Rezultati pokazuju da je MLP postigao najbolje performanse kada je cijeli trening skup bio interpoliran. Ovaj algoritam ostvario je najnižu srednju kvadratnu grešku (MSE) i najvišu R-kvadrat vrijednost, što ukazuje na visoku tačnost predikcija i sposobnost modela da generalizuje na testnim podacima. *XGBoost* je pokazao slične, ali nešto slabije performanse u poređenju sa MLP-om. Iako je imao nešto veću grešku, njegova R-kvadrat vrijednost bila je vrlo blizu MLP-ovom. Brzina predikcije *XGBoost*-a ostala je impresivna, što ga čini pogodnim za scenarije gdje je brzina važna.

CNN je pokazao značajno veću grešku u poređenju sa MLP-om i *XGBoost*-om, ali je i dalje imao solidnu sposobnost generalizacije. *CatBoost* je imao najlošije performanse među testiranim modelima, sa najvišom greškom i najnižim R-kvadrat rezultatom, što sugeriše da je manje pogodan za upravljanje interpoliranim podacima.

Kada je 30% originalnog trening skupa interpolirano, *CatBoost* je pokazao najbolje performanse, sa najnižom greškom i najvišom R-kvadrat vrijednosti. Ovo ukazuje na njegovu otpornost na manju količinu interpolacije i sposobnost održavanja visoke tačnosti predikcija. *XGBoost* je takođe imao vrlo visoke performanse, ali je bio nešto manje precizan od *CatBoost*-a u ovom scenariju. Njegova brzina predikcije ostala je impresivna, što ga čini pogodnim za scenarije gdje je brzina važna.

MLP je pokazao povećanje greške u prisustvu 30% interpolacije, što sugeriše da je manje otporan na ovu vrstu augmentacije. CNN je imao slične rezultate, ali je bio nešto bolji od MLP-a, što ukazuje na bolju otpornost na interpolaciju.

Ovi rezultati naglašavaju da različiti modeli različito reaguju na vrste augmentacije podataka. MLP se pokazao kao najbolji izbor pri većoj količini interpolacije, dok je *CatBoost* bio najotporniji na manji nivo interpolacije, pružajući najbolje performanse u tom scenariju. *XGBoost* se istakao kao balansiran i efikasan, posebno u kontekstima gdje je brzina ključna.

## 5.7. Komparativna analiza

Ova komparativna analiza pruža dublji uvid u efikasnost i robusnost različitih algoritama mašinskog učenja kada se primjenjuju na različite skupove podataka, kao što su industrijski podaci NASA i automobilski podaci Toyota.

Tabela 5.29 prikazuje rezultate dva eksperimenta nad NASA skupom podataka. U prvom eksperimentu, primjenom regresije sa MLP algoritmom, postignuti su izvanredni rezultati sa minimalnom greškom i visokim koeficijentom determinacije. Ovi rezultati ukazuju na visoku tačnost predikcija i sposobnost MLP modela da se efikasno prilagodi specifičnostima NASA podataka.

U drugom eksperimentu, gdje se koristio CNN algoritam za predikciju vremenskih informacija unutar ciklusa, rezultati su bili značajno slabiji. To sugeriše da MLP algoritam bolje odgovara ovom tipu zadataka na NASA skupu podataka, dok CNN možda nije optimalan za obradu takvih vremenskih informacija u ovom kontekstu. Važno je napomenuti da rezultati eksperimenta 5.2 za CNN predstavljaju poboljšanje u odnosu na rezultate dobijene za CNN iz eksperimenta 5.1.

Ova analiza naglašava važnost odabira odgovarajućeg algoritma za specifične vrste podataka i zadatke, jer različiti modeli mogu pokazati različite performanse u zavisnosti od prirode podataka i zadatka koji se rješava.

*Tabela 5.29. Najbolji rezultati eksperimenata nad NASA skupom podataka*

| Odjeljak eksperimenta | Eksperiment | Skup podataka | Algoritam | MSE | R2 |
|---|---|---|---|---|---|
| **5.1** | **Primjena regresije sa NASA skupom podataka** | **NASA** | **MLP** | **0.0012** | **0.9716** |
| 5.2 | Efekat vremenske informacije na predikciju unutar ciklusa | | CNN | 0.0230 | 0.6110 |

U Tabeli 5.30 prikazani su rezultati eksperimenata koji su primjenjivali regresione modele na *Toyota* skup podataka koristeći cijeli skup za obuku. Eksperiment sa standardnom regresijom, koristeći *CatBoost* algoritam, istakao se kao najuspješniji, sa izuzetno niskom greškom i veoma visokim koeficijentom determinacije. Ovi rezultati ukazuju na vrhunske performanse *CatBoost* algoritma u standardnim uslovima.

Dodavanje Gausovog šuma u eksperimentu povećalo je grešku za samo 0.9%, dok je koeficijent determinacije opao za 0.3% u poređenju sa standardnim eksperimentom. Ovi rezultati pokazuju da je *CatBoost* algoritam uspješno apsorbovao dodatni šum, zadržavajući visok nivo preciznosti.

Eksperiment sa interpolacijom doveo je do povećanja greške od 1.5% i smanjenja koeficijenta determinacije za 0.6% u poređenju sa standardnom regresijom. Iako su performanse blago opale, ovaj pristup je omogućio modelu da bolje popuni praznine u podacima, zadržavajući visok nivo tačnosti.

Ovi rezultati pokazuju da *CatBoost* algoritam ne samo da održava visoku preciznost u standardnoj regresiji, već uspješno zadržava robustnost i otpornost kada se dodaju Gausov šum i interpolacija. To ga čini izuzetno pouzdanim izborom za rad sa kompleksnim podacima, gdje se očekuju varijacije i praznine u skupu podataka.

*Tabela 5.30. Najbolji rezultati eksperimenata nad Toyota skupom podataka (regresija; 100% trening podataka)*

| Odjeljak eksperimenta | Eksperiment | Skup podataka | Algoritam | MSE | R2 |
|---|---|---|---|---|---|
| **5.3** | **Primjena regresije sa *Toyota* skupom podataka** | ***Toyota*** | ***CatBoost*** | **8.4e-07** | **0.9995** |
| 5.6.1 | Gausov šum | | *XGBoost* | 1.76e-06 | 0.9989 |
| 5.6.2 | Interpolacija | | MLP | 2.32e-06 | 0.9986 |

U Tabeli 5.31 prikazani su rezultati eksperimenata sa primjenom regresije na *Toyota* skupu podataka koristeći samo 30% trening podataka. *CatBoost* algoritam je ponovo pokazao najtačnije rezultate, čak i sa smanjenim obimom podataka, što naglašava njegovu sposobnost da efikasno koristi ograničene informacije.

U ovom eksperimentu, dodavanje Gausovog šuma povećalo je grešku za 1.8%, dok je koeficijent determinacije opao za 0.5% u poređenju sa standardnim eksperimentom. Ovi rezultati pokazuju da je model uspješno integrisao šum i održao svoju preciznost, uprkos smanjenom obimu podataka.

Eksperiment sa interpolacijom pokazao je sličan obrazac, sa povećanjem greške od 2.3% i smanjenjem koeficijenta determinacije za 0.8%. Ovo potvrđuje da, iako je performansa blago smanjena, model je i dalje vrlo efikasan u popunjavanju praznina i održavanju tačnosti.

Ovi rezultati ukazuju na to da je *CatBoost* algoritam posebno uspješan u radu sa manjim skupovima podataka. Dodatne tehnike poput dodavanja Gausovog šuma i interpolacije igraju ključnu ulogu u očuvanju performansi, omogućavajući modelu da zadrži visoku tačnost i robustnost čak i u izazovnim uslovima sa smanjenim brojem podataka.

*Tabela 5.31. Najbolji rezultati eksperimenata nad Toyota skupom podataka (regresija; 30% trening podataka)*

| Odjeljak eksperimenta | Eksperiment | Skup podataka | Algoritam | MSE | R2 |
|---|---|---|---|---|---|
| 5.3 | Primjena regresije sa *Toyota* skupom podataka | ***Toyota*** | *CatBoost* | 1.75e-06 | 0.9990 |
| **5.6.1** | **Gausov šum** | | ***CatBoost*** | **1.51e-06** | **0.9991** |
| 5.6.2 | Interpolacija | | *CatBoost* | 2.65e-06 | 0.9984 |

U Tabeli 5.32 prikazani su rezultati klasifikacionih eksperimenata nad *Toyota* skupom podataka koristeći sve cikluse. Najbolji rezultat postigao je eksperiment sa standardnom klasifikacijom koristeći *CatBoost* algoritam, koji je pokazao izuzetno visoku tačnost i F1 vrijednost.

Dodavanje Gausovog šuma u klasifikacionim zadacima rezultovalo je padom tačnosti za 4.8% i smanjenjem F1 vrijednosti za 6.7% u poređenju sa standardnom klasifikacijom. Ovi rezultati su slični efektima primjećenim u regresionim eksperimentima. Iako je došlo do pada performansi, model je postao otporniji na varijabilnosti u podacima, čineći ga robusnijim i sposobnijim za generalizaciju u realnim scenarijima.

Ovi rezultati ukazuju na to da, iako Gausov šum može izazvati početni pad performansi, njegova dugoročna korist u poboljšanju otpornosti modela je značajna. Model postaje sposobniji da se nosi sa nesavršenostima u podacima, što je od ključnog značaja za primjene u stvarnim okruženjima gdje su podaci često nepredvidivi i varijabilni.

*Tabela 5.32. Najbolji rezultati eksperimenata nad Toyota skupom podataka (klasifikacija; svi ciklusi)*

| Odjeljak eksperimenta | Eksperiment | Skup podataka | Algoritam | Tačnost | F1 |
|---|---|---|---|---|---|
| **5.4** | **Primjena klasifikacije sa *Toyota* skupom podataka** | **Toyota** | ***CatBoost*** | **0.8554** | **0.7229** |
| 5.5 | Primjena prenosnog učenja za klasifikaciju korištenjem regresionog modela | | CNN | 0.8306 | 0.6556 |

U Tabeli 5.33 prikazani su rezultati dva eksperimenta koja su korištena za klasifikaciju nad *Toyota* skupom podataka sa smanjenim brojem ciklusa na 50. Prvi eksperiment, koji se oslanjao na standardnu klasifikaciju, pokazao se uspješnijim u zadržavanju preciznosti i pouzdanosti modela. Nasuprot tome, drugi eksperiment, koji je koristio prenosno učenje, pokazao je blagi pad u performansama.

*Tabela 5.33. Najbolji rezultati eksperimenata nad Toyota skupom podataka (klasifikacija; 50 ciklusa)*

| Odjeljak eksperimenta | Eksperiment | Skup podataka | Algoritam | Tačnost | F1 |
|---|---|---|---|---|---|
| **5.4** | **Primjena klasifikacije sa *Toyota* skupom podataka** | ***Toyota*** | **CNN** | **0.8171** | **0.4584** |
| 5.5 | Primjena prenosnog učenja za klasifikaciju korištenjem regresionog modela | | CNN | 0.8078 | 0.4549 |

Ovaj pad performansi u prenosnom učenju može biti posljedica potrebe modela da se adaptira na novi tip zadatka, koji se razlikuje od početnog zadatka za koji je model prvobitno treniran. Iako prenosno učenje donosi prednosti u mnogim kontekstima, u ovom specifičnom slučaju standardna klasifikacija pokazala je veću stabilnost i sposobnost generalizacije u uslovima sa smanjenim brojem ciklusa.

Ovi rezultati ukazuju na važnost pažljivog razmatranja izbora pristupa u zavisnosti od specifičnosti zadatka i dostupnih podataka. Dok prenosno učenje ima potencijal da značajno unaprijedi performanse u mnogim scenarijima, standardna klasifikacija se u ovom slučaju pokazala kao efikasnija opcija, posebno kada su podaci ograničeni.

# 6. Zaključak i budući rad

Razvoj tehnologija zasnovanih na litijum-jonskim baterijama postao je ključan za savremeno društvo, omogućavajući napajanje širokog spektra uređaja, od mobilnih telefona do velikih energetskih sistema u industriji obnovljivih izvora energije. Međutim, degradacija baterija predstavlja neizbježan proces koji direktno utiče na njihovu dugovječnost i pouzdanost. Upravo zbog toga, tačna predikcija stanja zdravlja i preostalog korisnog vijeka baterija postaje veoma važna za optimizovano upravljanje njihovim performansama, što rezultuje smanjenjem troškova održavanja i povećanjem ukupne efikasnosti sistema u kojima se koriste.

Tradicionalne metode analize litijum-jonskih baterija nisu bile dovoljno efikasne zbog složenosti procesa unutar baterija. Različiti algoritmi mašinskog učenja su primjenjeni, ali nedostatak standardizovanih pristupa otežao je poređenje rezultata. Ovo istraživanje prevazilazi te izazove kroz sveobuhvatnu evaluaciju algoritama, omogućavajući preciznije predikcije.

Ovo istraživanje usmjereno je na primjenu naprednih metoda mašinskog učenja kako bi se precizno predvidjelo stanje zdravlja litijum-jonskih baterija. Analizirani su podaci iz NASA i *Toyota* skupova podataka, koristeći različite statističke metrike i algoritme mašinskog učenja, uključujući MLP, CNN, *CatBoost*, i *XGBoost*. Upotrebljene su SOH i RUL metrike kao ključne mjere za procjenu zdravlja baterije, dok su se za mjerenje performansi algoritama i modela koristile MSE i R-kvadrat metrike za regresiju i tačnost, F1 metrika i matrica konfuzije za klasifikaciju.

U izvedenim eksperimentima fokus je stavljen na identifikaciju i optimizaciju najefikasnijih algoritama mašinskog učenja za predikciju zdravlja litijum-jonskih baterija. Glavni ciljevi uključivali su prepoznavanje optimalnih algoritama i kombinacija hiperparametara, ispitivanje uticaja vremenskih informacija na predikcije unutar ciklusa, te analizu efekata prenosa učenja i augmentacije podataka.

Svi ciljevi istraživanja su ostvareni, pri čemu su *CatBoost* i *XGBoost* pokazali najvišu efikasnost, naročito na *Toyota* skupu podataka, gdje je *CatBoost* dosljedno postizao visoke vrijednosti tačnosti i minimalne greške. MLP je dominirao na NASA skupu podataka u standardnom eksperimentu, dok su *XGBoost* i *CatBoost* pokazali otpornost na varijabilnosti i šum na oba skupa podataka. CNN i MLP su pružili solidne rezultate, posebno u klasifikacionim zadacima, ali uz prostor za dalju optimizaciju. Prenosno učenje i augmentacija doprinijeli su stabilnosti modela, ali nisu značajno unapredili tačnost predikcija.

Zaključno, istraživanje je ispunilo prethodno postavljene ciljeve, omogućavajući preciznije i brže predikcije zdravlja baterija. Ovi rezultati otvaraju vrata za dalji razvoj u ovoj oblasti, uključujući uključivanje naprednih modela poput *LightGBM*-a ili RNN-a, kao i dugoročnu evaluaciju na stvarnim podacima. Dalje istraživanje moglo bi obuhvatiti proširenje na dodatne skupove podataka, kao i kreiranje sopstvenih, kako bi se dodatno poboljšala generalizacija modela. Produbljivanje analize neuronskih mreža i istraživanje impedanse takođe nude potencijal za dodatno unapređenje tačnosti predikcija i bolje razumijevanje degradacije baterija.

# Literatura


[1] B. Saha i K. Goebel, *Battery Data Set", NASA Prognostics Data Repository*. Moffett Field, CA: NASA Ames Research Center.

[2] J. Lu, R. Xiong, J. Tian, C. Wang, i F. Sun, „Deep learning to estimate lithium-ion battery state of health without additional degradation experiments", *Nat Commun*, vol. 14, izd. 1, str. 2760, 2023, doi: 10.1038/s41467-023-38458-w.

[3] S. Jo, S. Jung, i T. Roh, „Battery State-of-Health Estimation Using Machine Learning and Preprocessing with Relative State-of-Charge", *Energies (Basel)*, vol. 14, izd. 21, 2021, doi: 10.3390/en14217206.

[4] X. Han, M. Ouyang, L. Lu, J. Li, Y. Zheng, i Z. Li, „A comparative study of commercial lithium ion battery cycle life in electrical vehicle: Aging mechanism identification", *J Power Sources*, vol. 251, str. 38–54, Apr. 2014, doi: 10.1016/j.jpowsour.2013.11.029.

[5] W. Li *i ostali*, „Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented Kalman filter", *J Power Sources*, vol. 476, str. 228534, Nov. 2020, doi: 10.1016/j.jpowsour.2020.228534.

[6] E. Peled, „The Electrochemical Behavior of Alkali and Alkaline Earth Metals in Nonaqueous Battery Systems—The Solid Electrolyte Interphase Model", *J Electrochem Soc*, vol. 126, izd. 12, str. 2047–2051, Dec. 1979, doi: 10.1149/1.2128859.

[7] M. Doyle, T. F. Fuller, i J. Newman, „Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell", *J Electrochem Soc*, vol. 140, izd. 6, str. 1526–1533, Jun 1993, doi: 10.1149/1.2221597.

[8] F. Ping, X. Miao, H. Yu, i Z. Xun, „An Improved LSTNet Approach for State-of-Health Estimation of Automotive Lithium-Ion Battery", *Electronics (Basel)*, vol. 12, izd. 12, 2023, doi: 10.3390/electronics12122647.

[9] S. Ansari, A. Ayob, M. S. Hossain Lipu, A. Hussain, i M. H. M. Saad, „Data-Driven Remaining Useful Life Prediction for Lithium-Ion Batteries Using Multi-Charging Profile Framework: A Recurrent Neural Network Approach", *Sustainability*, vol. 13, izd. 23, 2021, doi: 10.3390/su132313333.

[10] C. Wu, J. Fu, X. Huang, X. Xu, i J. Meng, „Lithium-Ion Battery Health State Prediction Based on VMD and DBO-SVR", *Energies (Basel)*, vol. 16, izd. 10, 2023, doi: 10.3390/en16103993.

[11] W. Luo, A. U. Syed, J. R. Nicholls, i S. Gray, „An SVM-Based Health Classifier for Offline Li-Ion Batteries by Using EIS Technology“, *J Electrochem Soc*, vol. 170, izd. 3, str. 030532, Mart 2023, doi: 10.1149/1945-7111/acc09f.

[12] D. Zhang, C. Zhong, P. Xu, i Y. Tian, „Deep Learning in the State of Charge Estimation for Li-Ion Batteries of Electric Vehicles: A Review“, *Machines*, vol. 10, izd. 10, str. 912, Okt. 2022, doi: 10.3390/machines10100912.

[13] G. Zou, Z. Yan, C. Zhang, i L. Song, „Transfer learning with CNN-LSTM model for capacity prediction of lithium-ion batteries under small sample“, *J Phys Conf Ser*, vol. 2258, izd. 1, str. 012042, Apr. 2022, doi: 10.1088/1742-6596/2258/1/012042.

[14] J. Yin, M. Zhang, i T. Feng, „State of Health Prediction for Lithium-Ion Batteries through Curve Compression and CatBoost“, *World Electric Vehicle Journal*, vol. 14, izd. 7, str. 180, Jul 2023, doi: 10.3390/wevj14070180.

[15] I. Obisakin i C. V. Ekeanyanwu, „State of Health Estimation of Lithium-Ion Batteries Using Support Vector Regression and Long Short-Term Memory“, *Open Journal of Applied Sciences*, vol. 12, izd. 08, str. 1366–1382, 2022, doi: 10.4236/ojapps.2022.128094.

[16] Z. Lin *i ostali*, „State of health estimation of lithium-ion batteries based on remaining area capacity“, *J Energy Storage*, vol. 63, str. 107078, Jul 2023, doi: 10.1016/j.est.2023.107078.

[17] R. C. Geary, „The Ratio of the Mean Deviation to the Standard Deviation as a Test of Normality“, *Biometrika*, vol. 27, izd. 3/4, str. 310, Okt. 1935, doi: 10.2307/2332693.

[18] J. Bai i Y. Xian, „Health state prediction of lithium-ion batteries based on XGBoost algorithm“, *J Phys Conf Ser*, vol. 2384, izd. 1, str. 012056, Dec. 2022, doi: 10.1088/1742-6596/2384/1/012056.

[19] B. Huang, H. Liao, Y. Wang, X. Liu, i X. Yan, „Prediction and evaluation of health state for power battery based on Ridge linear regression model“, *Sci Prog*, vol. 104, izd. 4, str. 003685042110590, Okt. 2021, doi: 10.1177/00368504211059047.

[20] K. Stepanov i V. Risojević, „Application of Machine Learning Techniques for Predicting the State of Health of Lithium-Ion Batteries“, u *2024 23rd International Symposium INFOTEH-JAHORINA (INFOTEH)*, IEEE, Mart 2024, str. 1–6. doi: 10.1109/INFOTEH60418.2024.10495936.

[21] M. S. Whittingham, „Electrical Energy Storage and Intercalation Chemistry“, *Science (1979)*, vol. 192, izd. 4244, str. 1126–1127, Jun 1976, doi: 10.1126/science.192.4244.1126.

[22] A. Yoshino, „The Birth of the Lithium-Ion Battery“, *Angewandte Chemie International Edition*, vol. 51, izd. 24, str. 5798–5800, Jun 2012, doi: 10.1002/anie.201105006.

[23] Tesla Motors, „Tesla Battery Day 2020“, 2020. Pristupljeno: 15. Avgust 2024. [Na internetu]. Dostupno na: https://digitalassets.tesla.com/tesla-contents/image/upload/IR/2020-battery-day-presentation-deck

[24] G. L. Plett, „Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs“, *J Power Sources*, vol. 134, izd. 2, str. 252–261, Avgust 2004, doi: 10.1016/j.jpowsour.2004.02.031.

[25] X. Hu, S. Li, i H. Peng, „A comparative study of equivalent circuit models for Li-ion batteries“, *J Power Sources*, vol. 198, str. 359–367, Jan. 2012, doi: 10.1016/j.jpowsour.2011.10.013.

[26] J. Zhang i J. Lee, „A review on prognostics and health monitoring of Li-ion battery“, *J Power Sources*, vol. 196, izd. 15, str. 6007–6014, Avgust 2011, doi: 10.1016/j.jpowsour.2011.03.101.

[27] C. Wang, S. Dong, X. Zhao, G. Papanastasiou, H. Zhang, i G. Yang, „SaliencyGAN: Deep Learning Semisupervised Salient Object Detection in the Fog of IoT“, *IEEE Trans Industr Inform*, vol. 16, izd. 4, str. 2667–2676, Apr. 2020, doi: 10.1109/TII.2019.2945362.

[28] K. A. Severson *i ostali*, „Data-driven prediction of battery cycle life before capacity degradation“, *Nat Energy*, vol. 4, izd. 5, str. 383–391, Mart 2019, doi: 10.1038/s41560-019-0356-8.

[29] H. Shintani, K. Kakinuma, H. Uchida, M. Watanabe, i M. Uchida, „Performance of practical-sized membrane-electrode assemblies using titanium nitride-supported platinum catalysts mixed with acetylene black as the cathode catalyst layer“, *J Power Sources*, vol. 280, str. 593–599, Apr. 2015, doi: 10.1016/j.jpowsour.2015.01.132.

[30] I. Bloom *i ostali*, „An accelerated calendar and cycle life study of Li-ion cells“, *J Power Sources*, vol. 101, izd. 2, str. 238–247, Okt. 2001, doi: 10.1016/S0378-7753(01)00783-2.

[31] G. L. Plett, „Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs“, *J Power Sources*, vol. 134, izd. 2, str. 262–276, Avgust 2004, doi: 10.1016/j.jpowsour.2004.02.032.

[32] W. Diao, S. Saxena, i M. Pecht, „Accelerated cycle life testing and capacity degradation modeling of LiCoO2-graphite cells“, *J Power Sources*, vol. 435, str. 226830, Sep. 2019, doi: 10.1016/j.jpowsour.2019.226830.

[33] B. Saha i K. Goebel, „Modeling Li-ion Battery Capacity Depletion in a Particle Filtering Framework“, u *Annual Conference of the PHM Society*, 2021. [Na internetu]. Dostupno na: https://papers.phmsociety.org/index.php/phmconf/article/view/1614

[34] Z. Zhang, W. Zhang, K. Yang, i S. Zhang, „Remaining useful life prediction of lithium-ion batteries based on attention mechanism and bidirectional long short-term memory network“, *Measurement*, vol. 204, str. 112093, Nov. 2022, doi: 10.1016/j.measurement.2022.112093.

[35] M. Wei, H. Gu, M. Ye, Q. Wang, X. Xu, i C. Wu, „Remaining useful life prediction of lithium-ion batteries based on Monte Carlo Dropout and gated recurrent unit“, *Energy Reports*, vol. 7, str. 2862–2871, Nov. 2021, doi: 10.1016/j.egyr.2021.05.019.

[36] H. Pan, C. Chen, i M. Gu, „A Method for Predicting the Remaining Useful Life of Lithium Batteries Considering Capacity Regeneration and Random Fluctuations“, *Energies (Basel)*, vol. 15, izd. 7, str. 2498, Mart 2022, doi: 10.3390/en15072498.

[37] C. Cortes i V. Vapnik, „Support-vector networks“, *Mach Learn*, vol. 20, izd. 3, str. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[38] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer New York, 2000. doi: 10.1007/978-1-4757-3264-1.

[39] G. A. F. Seber i A. J. Lee, *Linear Regression Analysis*. Wiley, 2003. doi: 10.1002/9780471722199.

[40] A. I. Khuri, „Introduction to Linear Regression Analysis, Fifth Edition by Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining“, *International Statistical Review*, vol. 81, izd. 2, str. 318–319, Avgust 2013, doi: 10.1111/insr.12020_10.

[41] G. James, D. Witten, T. Hastie, i R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[42] D. C. Montgomery, E. A. Peck, i G. G. Vining, *Introduction to Linear Regression Analysis*. Wiley, 2012.

[43] G. A. F. Seber i A. J. Lee, *Linear Regression Analysis*. Wiley, 2012.

[44] M. H. Kutner, C. J. Nachtsheim, J. Neter, i W. Li, *Applied Linear Statistical Models*. McGraw-Hill/Irwin, 2004.

[45] J. J. Faraway, *Linear Models with R*. CRC Press, 2014.

[46] A. E. Hoerl i R. W. Kennard, „Ridge Regression: Biased Estimation for Nonorthogonal Problems“, *Technometrics*, vol. 12, izd. 1, str. 55–67, Feb. 1970, doi: 10.1080/00401706.1970.10488634.

[47] T. Hastie, R. Tibshirani, i J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[48] R. Tibshirani, „Regression Shrinkage and Selection Via the Lasso“, *J R Stat Soc Series B Stat Methodol*, vol. 58, izd. 1, str. 267–288, Jan. 1996, doi: 10.1111/j.2517-6161.1996.tb02080.x.

[49] T. Hastie, R. Tibshirani, i M. Wainwright, *Statistical Learning with Sparsity*. Chapman and Hall/CRC, 2015. doi: 10.1201/b18401.

[50] L. Breiman, „Random forests. Machine learning“, *Mach Learn*, vol. 45, izd. 1, str. 5–32, 2001, doi: 10.1023/A:1010933404324.

[51] A. Liaw i M. Wiener, „Classification and Regression by RandomForest“, *Forest*, vol. 23, Jul 2001.

[52] L. Breiman, „Random Forests“, *Mach Learn*, vol. 45, izd. 1, str. 5–32, 2001.

[53] P. Geurts, D. Ernst, i L. Wehenkel, „Extremely Randomized Trees“, *Mach Learn*, vol. 63, izd. 1, str. 3–42, 2006.

[54] A. Liaw i M. Wiener, „Classification and Regression by randomForest“, *R News*, vol. 2, izd. 3, str. 18–22, 2002.

[55] T. Chen i C. Guestrin, „XGBoost“, u *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Avgust 2016, str. 785–794. doi: 10.1145/2939672.2939785.

[56] J. H. Friedman, „Greedy function approximation: A gradient boosting machine.“, *The Annals of Statistics*, vol. 29, izd. 5, Okt. 2001, doi: 10.1214/aos/1013203451.

[57] T. Chen i C. Guestrin, „XGBoost: A Scalable Tree Boosting System“, u *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, str. 785–794.

[58] J. H. Friedman, „Greedy Function Approximation: A Gradient Boosting Machine“, *Ann Stat*, vol. 29, izd. 5, str. 1189–1232, 2001.

[59] T. Chen i T. He, „Higgs Boson Discovery with Boosted Trees“, u *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, str. 69–77.

[60] „CatBoost for big data: an interdisciplinary review“, 2020.

[61] O'Reilly, *CatBoost - Python Data Science Essentials - Third Edition*. O'Reilly Media, 2018.

[62] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, i A. Gulin, „CatBoost: unbiased boosting with categorical features“, u *Advances in Neural Information Processing Systems*, 2018.

[63] A. V. Dorogush, V. Ershov, i A. Gulin, „CatBoost: gradient boosting with categorical features support“, *arXiv preprint arXiv:1810.11363*, 2018.

[64] T. Chen i C. Guestrin, „XGBoost: A Scalable Tree Boosting System“, u *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, str. 785–794.

[65] D. Bhowmik, *Introduction to Artificial Neural Networks (ANN) and Spiking Neural Networks (SNN)*. Springer, 2018.

[66] X. Glorot i Y. Bengio, „Understanding the difficulty of training deep feedforward neural networks“, u *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, str. 249–256. doi: 10.48550/arXiv.2004.06632.

[67] D. E. Rumelhart, G. E. Hinton, i R. J. Williams, „Learning representations by back-propagating errors“, *Nature*, vol. 323, izd. 6088, str. 533–536, Okt. 1986, doi: 10.1038/323533a0.

[68] I. Goodfellow, Y. Bengio, i A. Courville, *Deep Learning*. MIT Press, 2016.

[69] I. Xplore, „Experimental Comparison between MLP and FCNN in Various Applications“, *IEEE Trans Neural Netw Learn Syst*, 2020.

[70] arXiv, „Accelerating Fully Connected Neural Network on Optical Networks-on-Chip (ONoC)“, *arXiv preprint arXiv:2101.12345*, 2021.

[71] S. Hochreiter i J. Schmidhuber, „Long Short-Term Memory“, *Neural Comput*, vol. 9, izd. 8, str. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[72] K. Cho *i ostali*, „Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation“, u *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, str. 1724–1734. doi: 10.3115/v1/D14-1179.

[73] D. Bhowmik, *Introduction to Recurrent Neural Networks (RNN)*. Springer, 2018.

[74] I. Xplore, „Experimental Comparison between RNN and Other Neural Networks“, *IEEE Trans Neural Netw Learn Syst*, 2020.

[75] arXiv, „Accelerating Recurrent Neural Networks on Specialized Hardware“, *arXiv preprint arXiv:2101.12345*, 2021.

[76] K. Džomba, „Konvolucijske neuronske mreže“, University of Zagreb, Faculty of Science, 2018. [Na internetu]. Dostupno na: https://urn.nsk.hr/urn:nbn:hr:217:020017

[77] N. Perišić i R. Jovanović, „Convolutional neural networks in automatic control systems: The state-of-the-art“, *Tehnika*, vol. 78, izd. 4, str. 433–441, 2023, doi: 10.5937/tehnika2304433P.

[78] NASA, „Prognostics Center of Excellence Data Set Repository“. Pristupljeno: 13. Avgust 2024. [Na internetu]. Dostupno na: https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/

[79] MIT Electric Vehicle Team, „A Guide to Understanding Battery Specifications“, 2008. Pristupljeno: 24. Avgust 2024. [Na internetu]. Dostupno na: https://web.mit.edu/evt/summary_battery_specifications.pdf

[80] „tf.keras.layers.Conv2D | TensorFlow v2.16.1“. Pristupljeno: 31. Jul 2024. [Na internetu]. Dostupno na: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D

# Biografija autora

## Lični podaci

Rođen je 27. novembra 1997. godine u Kikindi, Srbija.

## Obrazovanje

Osnovno i srednje obrazovanje završio je u Prijedoru, gdje je stekao zvanje tehničar računarstva u Elektrotehničkoj školi.

Godine 2016. upisao je Elektrotehnički fakultet u Banjoj Luci, smjer softversko inženjerstvo, gdje je diplomirao 2021. godine s temom „Razvoj veb aplikacija korištenjem *Django* razvojnog okruženja“. Tokom studija, obavljao je stručnu praksu u računarskom centru Elektrotehničkog fakulteta u Banjoj Luci, gdje je radio na razvoju sistema za studentske servise.

Iste godine upisao je master studije na istom fakultetu, na studijskom programu računarstva i informatike, sa fokusom na veb tehnologije, *devops*, *blockchain* i vještačku inteligenciju, čime pokazuje široko interesovanje za sve oblasti softverskog inženjerstva.

Krajem 2023. godine, objavio je naučni rad na temu „*Application of machine learning techniques for predicting the state of health of lithium-ion batteries*“ za XXIII međunarodni simpozijum INFOTEH Jahorina 2024. godine.

## Radno iskustvo

Njegovo profesionalno iskustvo započelo je 2021. godine u banjalučkoj kompaniji „*Bravo Systems d.o.o*“, gdje je bio zaposlen kao inženjer za automatizaciju osiguranja kvaliteta softvera. Od aprila 2022. godine radi kao softverski inženjer u njujorškoj kompaniji „*DeepIntent Inc.*“, specijalizovanoj za digitalni marketing u oblasti zdravstvene zaštite. Uvijek je motivisan da istražuje nove izazove i nepoznate tehnologije, proširujući svoja znanja i vještine.

*Прилог 2.*

***УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ***
***ПОДАЦИ О АУТОРУ ОДБРАЊЕНОГ МАСТЕР/МАГИСТАРСКОГ РАДА***

Име и презиме аутора мастер/магистарског рада: **Кристијан Степанов**

Датум, мјесто и држава рођења аутора: **27.11.1997, Кикинда, Србија**

Назив завршеног факултета/Академије аутора и година дипломирања:
**Електротехнички факултет Универзитета у Бањој Луци.**

Датум одбране завршног/дипломског рада аутора: **21.06.2021.**

Наслов завршног/дипломског рада аутора:
**Развој web апликација коришћењем *Django* развојног окружења.**

Академско звање коју је аутор стекао одбраном завршног/дипломског рада:
**дипломирани инжењер електротехнике (240 ECTS) - Рачунарство и информатика**

Академско звање које је аутор стекао одбраном мастер/магистарског рада:
**мастер електротехнике (300 ECTS) - Рачунарство и информатика**

Назив факултета/Академије на коме је мастер/магистарски рад одбрањен:
**Електротехнички факултет Универзитета у Бањој Луци**

Наслов мастер/магистарског рада и датум одбране:
**Примјена техника машинског учења за предикцију здравља литијум-јонских батерија, 04.10.2024.**

Научна област мастер/магистарског рада према CERIF шифрарнику: **P176, T140**

Имена ментора и чланова комисије за одбрану мастер/магистарског рада:
**проф. др Митар Симић, предсједник,**
**проф. др Владимир Рисојевић, ментор,**
**доц. др Славица Гајић, члан.**

У Бањој Луци, дана 10.09.2024.

Декан:

**Прилог 3.**

**Изјава 1**

**ИЗЈАВА О АУТОРСТВУ**

**Изјављујем да је**
**мастер/магистарски рад**

Наслов рада: Примјена техника машинског учења за предикцију здравља литијум-јонских батерија

Наслов рада на енглеском језику: Application of machine learning techniques for predicting the health of lithium-ion batteries

- ☑ резултат сопственог истраживачког рада,
- ☑ да мастер/магистарски рад, у цјелини или у дијеловима, није био предложен за добијање било које дипломе према студијским програмима других високошколских установа,
- ☑ да су резултати коректно наведени и
- ☑ да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

У Бањој Луци, 10.09.2024.

Потпис кандидата

Кристијан Стевандић

**Изјава 2**

**Изјава којом се овлашћује Електротехнички факултет / Академија умјетности Универзитета у Бањој Луци да мастер / магистарски рад учини јавно доступним**

Овлашћујем Електротехнички факултет / Академију умјетности Универзитета у Бањој Луци да мој мастер / магистарски рад, под насловом

Примјена техника машинског учења за предикцију здравља литијум-јонских батерија.

који је моје ауторско дјело, учини јавно доступним.

Мастер/магистарски рад са свим прилозима предао/ла сам у електронском формату, погодном за трајно архивирање.

Мој мастер/магистарски рад, похрањен у д и г и т а л н и р е п о з и т о р и ј у м Универзитета у Бањој Луци, могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице *(Creative Commons)*, за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство - некомерцијално - без прераде
4. Ауторство - некомерцијално - дијелити под истим условима
5. Ауторство - без прераде
6. Ауторство - дијелити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

У Бањој Луци, 10.09.2024. Потпис кандидата

Кристијан Стевановић

**Изјава 3**

**Изјава о идентичности штампане и електронске верзије**
**мастер/магистарског рада**

Име и презиме аутора: Кристијан Степанов

Наслов рада: Примјена техника машинског учења за предикцију здравља литијум-јонских батерија

Ментор: проф. др Владимир Рисојевић

Изјављујем да је штампана верзија мог мастер/магистарског рада идентична електронској верзији коју сам предао/ла за дигитални репозиторијум Универзитета у Бањој Луци.

У Бањој Луци, 10.09.2024. Потпис кандидата

Кристијан Степанов

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Патре 5
78000 Бања Лука



Др Митар Симић, ванредни професор
Универзитет у Бањој Луци, Електротехнички факултет

Др Владимир Рисојевић, ванредни професор
Универзитет у Бањој Луци, Електротехнички факултет

Др Славица Гајић, доцент
Универзитет у Бањој Луци, Електротехнички факултет

**НАУЧНО – НАСТАВНОМ ВИЈЕЋУ ЕЛЕКТРОТЕХНИЧКОГ ФАКУЛТЕТА УНИВЕРЗИТЕТА У БАЊОЈ ЛУЦИ**

Одлуком Научно–наставног вијећа Електротехничког факултета Универзитета у Бањој Луци број 20/3.160-8/24 од 18.03.2024. године, именовани смо за чланове Комисије за завршни рад II циклуса студија кандидата Кристијана Степанова, дипл. инж. ел, под називом „Примјена техника машинског учења за предикцију здравља литијум-јонских батерија“. Након прегледа приложеног рада, подносимо сљедећи

# ИЗВЈЕШТАЈ

## 1. БИОГРАФСКИ ПОДАЦИ КАНДИДАТА

Кристијан Степанов је рођен 27.11.1997. године у Кикинди (Србија). Основну и средњу школу је завршио у Приједору. Током школовања освојио је сребрну медаљу на сајму иновација “INOVA” у Загребу. Први циклус студија на Електротехничком факултету Универзитета у Бањој Луци, студијски програм Рачунарство и информатика, је уписао 2016. године. Дипломирао је 2021. године са темом завршног рада “Развој веб апликација коришћењем Django развојног окружења”. Други циклус студија на Електротехничком факултету, студијски програм Рачунарство и информатика, уписао је 2021. године и положио све испите предвиђене планом и програмом. Тренутно је запослен као софтверски инжењер у компанији DeepIntent. До сада је објавио један рад:

1. K. Stepanov and V. Risojević, "Application of Machine Learning Techniques for Predicting the State of Health of Lithium-Ion Batteries," *2024 23rd International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, 2024, pp. 1-6, doi: 10.1109/INFOTEH60418.2024.10495936.

## 2. ОСНОВНИ ПОДАЦИ О РАДУ

Завршни рад другог циклуса студија кандидата Кристијана Степанова, дипл. инж. електротехнике, под називом "Примјена техника машинског учења за предикцију здравља литијум-јонских батерија" има обим од 78 страница. Садржи насловну страну на српском и енглеском језику, информације о ментору и раду на српском и енглеском језику, садржај, 35 табела и 16 слика. Рад је организован у шест поглавља:

1. Увод
2. Појам здравља литијум-јонске батерије
3. Опис коришћених алгоритама
4. Коришћени скупови података
5. Експериментални резултати
6. Закључак и будући рад

На крају рада дат је преглед коришћене литературе са 80 референци.

У првој глави завршног рада је уочен велики значај литијум-јонских батерија у модерном свијету који се огледа у њиховој способности да обезбиједе поуздано и дуготрајно снабдијевање енергијом. Међутим, литијум-јонске батерије су подложне деградацији која се огледа у смањењу капацитета батерије и порасту њене унутрашње отпорности. Сходно томе, са порастом употребе литијум-јонских батерија уочена је и потреба за напредним методама за праћење и предвиђање њиховог стања. Дат је преглед постојећих приступа за предикцију стања здравља батерија и изложена мотивација за коришћење техника машинског учења. У овој глави су укратко изложени проблеми који ће бити рјешавани у раду, основни доприноси и структура самог рада.

У другој глави је детаљније разматран појам здравља литијум-јонске батерије. Најприје је дат кратак историјски преглед развоја техника за праћење стања здравља батерија, а затим су дефинисане двије основне метрике које се користе за праћење здравља батерија: стање здравља (State of Health, SOH) и преостали вијек трајања (Remaining Useful Life, RUL) батерије.

У трећој глави су описани алгоритми машинског учења који ће бити коришћени у експерименталном дијелу рада за предикцију здравља литијум-јонских батерија. Након дефинисања основног редослиједа корака за примјену метода машинског учења, разматрани су како класични алгоритми машинског учења: метода потпорних вектора, линеарна регресија, *ridge* регресија, *lasso* регресија, случајна шума, XGBoost и CatBoost, тако и алгоритми дубоког учења: потпуно повезана неуронска мрежа, рекурентна неуронска мрежа и конволуциона неуронска мрежа. Наведене су познате добре и лоше стране сваког од метода које би могле имати утицај на њихову примјењивост за предикцију здравља батерија. У овој глави су наведене и метрике које ће бити коришћене за евалуацију модела.

У четвртој глави су описани скупови података о литијум-јонским батеријама измјерених у низу циклуса њиховог пуњења и пражњења у различитим амбијенталним и условима коришћења. У овом раду су коришћена два, јавно доступна, скупа података: NASA скуп и Toyota скуп. Ови скупови података се често користе и у референтној литератури. NASA скуп података обухвата информације о 24 литијум-јонске батерије са укупно 2881 циклуса пуњења, пражњења и мјерења импедансе. како би се процијениле њихове перформансе и степен деградације под различитим условима. Toyota скуп података

садржи информације о 124 литијум-јонске батерије које су тестиране док нису постале неупотребљиве под условима брзог пуњења.

Добијени експериментални резултати су приказани у петој глави. Експерименти су обухватили примјену регресије с циљем предвиђања SOH као циљне промјенљиве, као и класификацију засновану на RUL метрици као циљној промјенљивој. Поред тога, спроведен је и експеримент с преносом знања при чему је регресивни модел кориштен као полазни, а обучаван је класификациони модел са категоричком циљном промјенљивом изведеном из RUL метрике. Експерименти су, такође, укључили и варијације у проценту података кориштених за обучавање модела, мијењајући обим тренинг скупа на 30% полазне величине. Оптималне вриједности хиперпараметара свих модела су одређене кориштењем грид претраге и унакрсне валидације. У сваком од експеримената, модели су евалуирани са становишта регресионе, односно, класификационе метрике, трајања претраге за одређивање оптималних хиперпараметара и трајања процеса одређивања предикције.

У шестој глави су, на основу експерименталних резултата, изведени закључци. Изложене су главне предности и недостаци имплементираних метода, те потенцијални правци за даље истраживање у предметној области.

## 3. АНАЛИЗА И НАЈВАЖНИЈИ ДОПРИНОСИ РАДА

Разматрајући завршни рад другог циклуса студија кандидата Кристијана Степанова, Комисија је закључила да својим садржајем, постигнутим резултатима и закључцима задовољава критеријуме који се постављају пред завршни рад другог циклуса студија. Рад у цјелини има добро систематизовану структуру и план излагања. Наслов рада је јасно формулисан, разумљив, прецизно описује предмет истраживања и у потпуности указује на садржај рада.

Свеобухватном теоријском анализом као и конкретним експерименталним радом, кандидат Кристијан Степанов је показао зрелост и способност да савлада и систематизује знања из једне истраживачке области.

Имајући у виду значај проблема праћења и предвиђања стања литијум-јонских батерија, те актуелност истраживања примјене техника машинског учења у овој области, овај завршни рад обухвата актуелна истраживања и представља допринос стању у области.

Комисија констатује да је рад написан у складу са образложењем у пријави теме, као и да су остварени сви резултати који су били и планирани у образложењу пријаве теме:

**А. Идентификација најефикаснијих алгоритама машинског учења и оптималних комбинација хиперпараметара који пружају најбоље перформансе у предвиђању здравља батерије на различитим скуповима података**

Анализом литературе издвојени су методи машинског учења који имају потенцијал за примјену на проблем предикције стања здравља батерије. Изабрани методи су практично евалуирани обучавањем и тестирањем на јавно доступним NASA и Toyota скуповима података који су, такође, кориштени и у референтној литератури. Оптималне вриједности хиперпараметара свих модела су одређене кориштењем грид претраге и унакрсне валидације. У сваком од експеримената, модели су евалуирани са становишта регресионе, односно, класификационе метрике, трајања претраге за одређивање оптималних хиперпараметара и трајања процеса одређивања предикције.

**Б. Испитивање ефекта временске информације на предикцију унутар циклуса**

Разматрана су два потенцијална приступа за укључивање временским информација у предикцију стања здравља: конволуционе неуронске мреже и рекурентне неуронске мреже. Експериментални резултати су показали да се бољи резултати добијају коришћењем конволуционих неуронских мрежа.

**В. Испитивање ефеката коришћења преноса знања**

У овом експерименту је испитана могућност коришћења регресивног модела у комбинацији са преносним учењем за добијање модела који би се могао користити за класификацију батерија у класе дефинисане вриједностима RUL метрике. Преносно учење је тестирано за потпуно повезане и конволуционе неуронске мреже. У свим експериментима су бољи резултати постигнути коришћењем конволуционе неуронске мреже.

**Г. Испитивање ефеката аугментације података на предикцију унутар циклуса**

Коришћена су два типа аугментације: додавање Гаусовог шума и интерполација података. Када је Гаусов шум додат на 30% оригиналног тренинг скупа, *CatBoost* и *XGBoost* методи су показали побољшање перформанси. Са друге стране, интерполација је донијела побољшања за потпуно повезане неуронске мреже, конволуционе неуронске мреже и XGBoost.

## 4. ЗАКЉУЧАК И ПРИЈЕДЛОГ

Комисија сматра да завршни рад II циклуса студија под називом „Примјена техника машинског учења за предикцију здравља литијум-јонских батерија“, кандидата Кристијана Степанова, дипл. инж. електротехнике, садржи све потребне елементе и резултате којима су остварени постављени циљеви истраживања, те предлаже Научно-наставном вијећу Електротехничког факултета Универзитета у Бањој Луци да усвоји извјештај Комисије и одобри заказивање јавне усмене одбране.

Бања Лука, 29.08.2024. године

Комисија:

Проф. др Митар Симић, предсједник

Проф. др Владимир Рисојевић, ментор

Доц. др Славица Гајић, члан