

Predikcija postupaka održavanja uljem uronjenih transformatora na temelju fizikalno-kemijskih analiza

Delić, Igor

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:089147>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-19**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

PREDIKCIJA POSTUPAKA ODRŽAVANJA ULJEM
URONJENIH TRANSFORMATORA NA TEMELJU
FIZIKALNO-KEMIJSKIH ANALIZA

Diplomski rad

Igor Delić

Osijek, 2024.

Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju

Ocjena diplomskog rada na sveučilišnom diplomskom studiju

Ime i prezime pristupnika:	Igor Delić
Studij, smjer:	Sveučilišni diplomski studij Računarstvo
Mat. br. pristupnika, god.	D1278R, 07.10.2022.
JMBAG:	0165081464
Mentor:	izv. prof. dr. sc. Alfonzo Baumgartner
Sumentor:	
Sumentor iz tvrtke:	Veljko Špica
Predsjednik Povjerenstva:	izv. prof. dr. sc. Tomislav Keser
Član Povjerenstva 1:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 2:	doc. dr. sc. Tomislav Galba
Naslov diplomskog rada:	Predikcija postupaka održavanja uljem uronjenih transformatora na temelju fizikalno-kemijskih analiza
Znanstvena grana diplomskog rada:	Obradba informacija (zn. polje računarstvo)
Zadatak diplomskog rada:	[Rezervirano: Igor Delić] Na osnovu povijesnih podataka fizikalno-kemijskih analiza koji su arhivirani u bazi, potrebno je metodama strojnog učenja izraditi model klasifikacije koji će ocijeniti stanje transformatora na osnovu izmjerenih parametara. Izraditi odgovarajuću aplikaciju koja će omogućiti unos parametara, izračunati i ispisati indeks zdravlja ("health index") transformatora te uputiti u postupke održavanja s obzirom na zadanu kategorizaciju. Pri izradi modela strojnog učenja potrebno je ukloniti sve grube i sistemske
Datum ocjene pismenog dijela diplomskog rada od strane mentora:	05.07.2024.
Ocjena pismenog dijela diplomskog rada od strane mentora:	Izvrstan (5)
Datum obrane diplomskog rada:	16.7.2024
Ocjena usmenog dijela diplomskog rada (obrane):	Izvrstan (5)
Ukupna ocjena diplomskog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:	16.07.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 16.07.2024.

Ime i prezime Pristupnika:

Igor Delić

Studij:

Sveučilišni diplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

D1278R, 07.10.2022.

Turnitin podudaranje [%]:

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Predikcija postupaka održavanja uljem uronjenih transformatora na temelju fizikalno-kemijskih analiza**

izrađen pod vodstvom mentora izv. prof. dr. sc. Alfonzo Baumgartner

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak diplomskog rada.....	1
2. TRANSFORMATORI I NJIHOVA ULOGA.....	2
2.2. Ključne komponente i vrste	3
2.3. Praćenje stanja transformatora.....	5
2.4. Indeks zdravlja transformatora	8
2.5. Metode procjene i održavanja kvalitete ulja.....	10
3. RAZVOJ MODELA	13
3.1. Strojno učenje	13
3.1.1. Nadzirano strojno učenje.....	14
3.1.2. Nenadzirano strojno učenje	15
3.1.3. Podržano strojno učenje	15
3.2. Analiza i podjela podataka	16
3.3. Usporeba i odabir algoritma za izradu modela.....	21
3.3.1. RandomForestClassifier	21
3.3.2. Support Vector Machine	22
3.3.3. K-Nearest Neighbors.....	23
3.3.4. Rezultati usporedbe modela	24
4. IZRADA APLIKACIJE	25
4.1. Treniranje i pohrana modela.....	25
4.2. Streamlit aplikacija	27
ZAKLJUČAK	35
LITERATURA.....	36
SAŽETAK.....	37
ABSTRACT	38

1. UVOD

Transformatori su jako bitne komponente elektroenergetskih sustava i imaju bitnu ulogu u prijenosu električne energije. Njihova pouzdanost utječe na stabilnost i učinkovitost cijele mreže., zato je bitno konstantno pratiti i procjenjivati stanje transformatora kako bi se osigurala pravovremena reakcija i zaustavili kvarovi koji mogu uzrokovati ozbiljne smetnje i financijske gubitke. Ovaj diplomski rad ima za cilj razviti model za ocjenu stanja uljem uronjenih transformatora korištenjem metoda strojnog učenja, na temelju povijesnih podataka fizikalno-kemijskih analiza ulja pohranjenih u bazi podataka. Analizom i interpretacijom ovih podataka omogućit će se preciznija i učinkovitija procjena stanja transformatora. Primjenom strojnog učenja i samog modela ovaj će rad omogućiti predviđanje stanja ulja transformatora na temelju izmjerenih parametara, što će pomoći u prepoznavanju potencijalnih problema i održavanju. Glavni dio ovog rada je razvoj aplikacije koja će korisnicima omogućiti unos parametara, izvođenje potrebnih izračuna i prikaz rezultata u obliku trenutnog stanja ulja transformatora.

Nakon samog uvoda u prvom poglavlju, u drugom poglavlju se opisuje rad transformatora i njegova uloga te zašto je važno pratiti stanje transformatora i redovno održavati. U trećem poglavlju se definiraju pojmovi vezani uz strojno učenje, analiziraju se i pripremaju podaci za trening te se odabire najbolji model za nas skup podataka. U četvrtom poglavlju ovog rada prolazi se kroz proces izrade Streamlit aplikacije i opisuje se način rada aplikacije.

1.1.Zadatak diplomskog rada

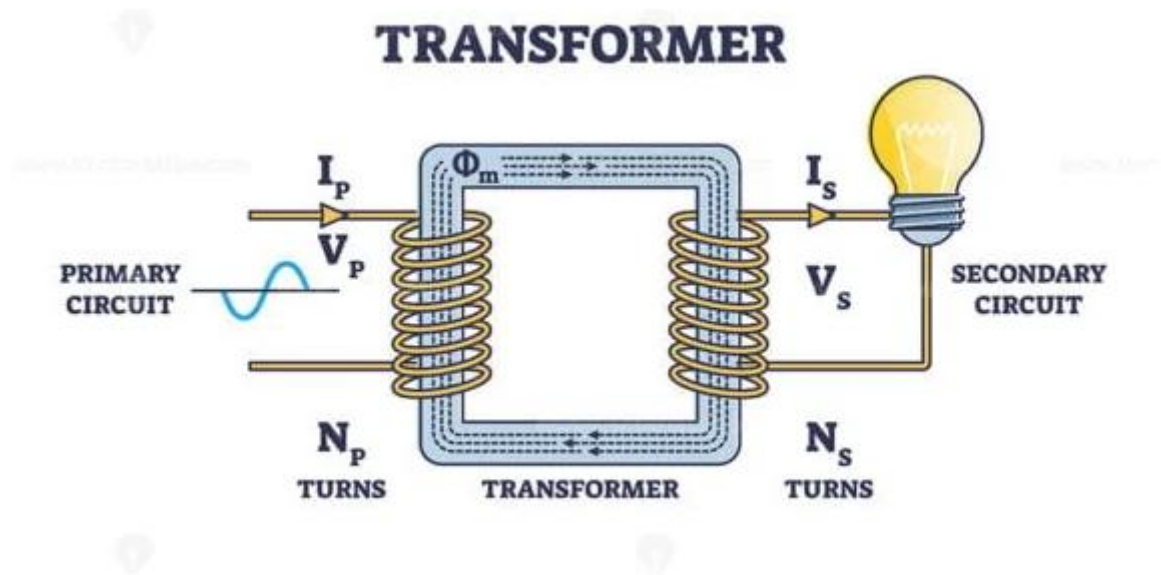
Na osnovu povijesnih podataka fizikalno-kemijskih analiza koji su arhivirani u bazi, potrebno je metodama strojnog učenja izraditi model klasifikacije koji će ocijeniti stanje transformatora na osnovu izmjerenih parametara. Izraditi odgovarajuću aplikaciju koja će omogućiti unos parametara, izračunati i kvalitetu ulja transformatora te uputiti u postupke održavanja s obzirom na zadanu kategorizaciju. Pri izradi modela strojnog učenja potrebno je ukloniti sve grube i systemske pogreške, te ugraditi predikciju nepostojećih podataka.

2. TRANSFORMATORI I NJIHOVA ULOGA

U ovom poglavlju se prolazi kroz teorijsku pozadinu samog rada transformatora, koje su ključne komponente i vrste te kako i zašto je potrebno pratiti index zdravlja transformatora. Također se upoznaje s metodama procjene kvalitete i metodama održavanja ulja.

2.1. Princip rada

Transformatori su bitne komponente u distribuciji električne energije. Rade na osnovnom principu elektromagnetske indukcije, prenoseći električnu energiju između dva ili više krugova putem elektromagnetske veze.



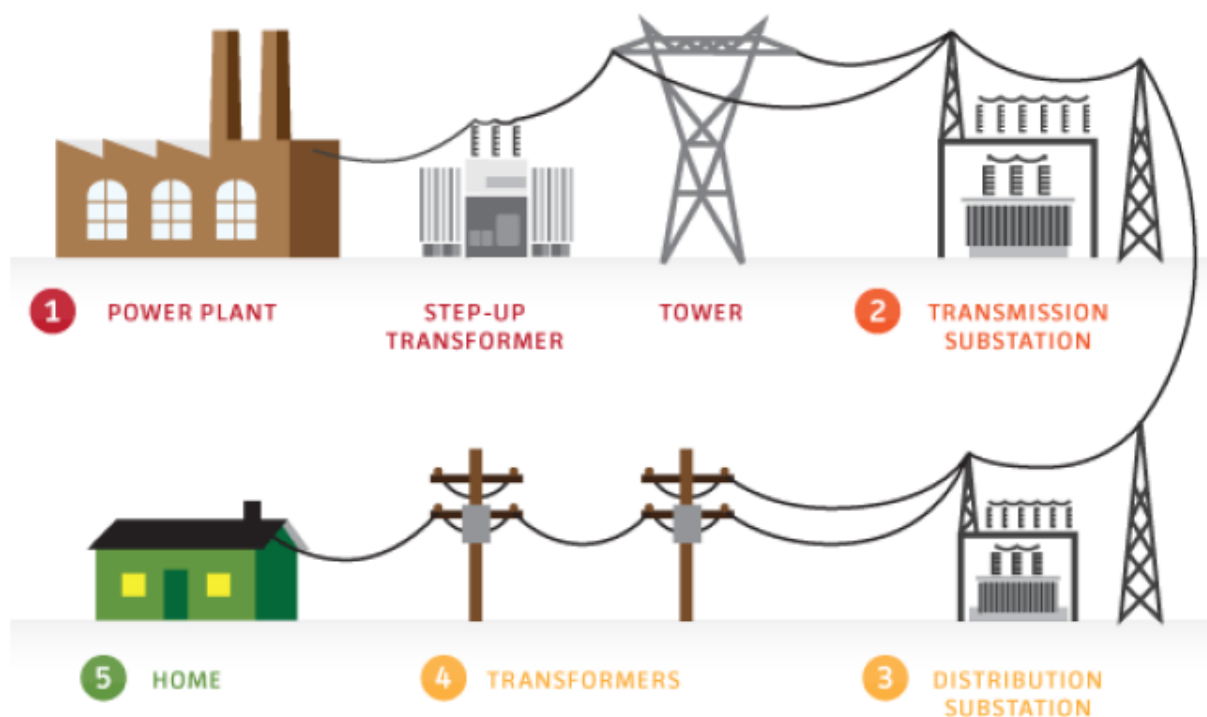
Slika 2.1. Prikaz principa rada transformatora.

Transformator se sastoji od primarnog i sekundarnog namota omotanih oko magnetske jezgre. Kada izmjenična struja teče kroz primarni namot, ona stvara promjenjivo magnetsko polje u jezgri koje inducira napon u sekundarnom namotu, prenoseći tako energiju iz primarnog u sekundarni krug.

2.2. Ključne komponente i vrste

Osnovne komponente transformatora su namoti koji se sastoje od zavojnica i zica kroz koje teče struja i jezgra koja je najčešće izrađena od čelika i povećava magnetski tok i učinkovitost transformatora. Od ostalih komponenti bitni su izolacija za održavanje sigurnosti te sustavi hlađenja poput ulja za raspršivanje topline nastale radom transformatora.

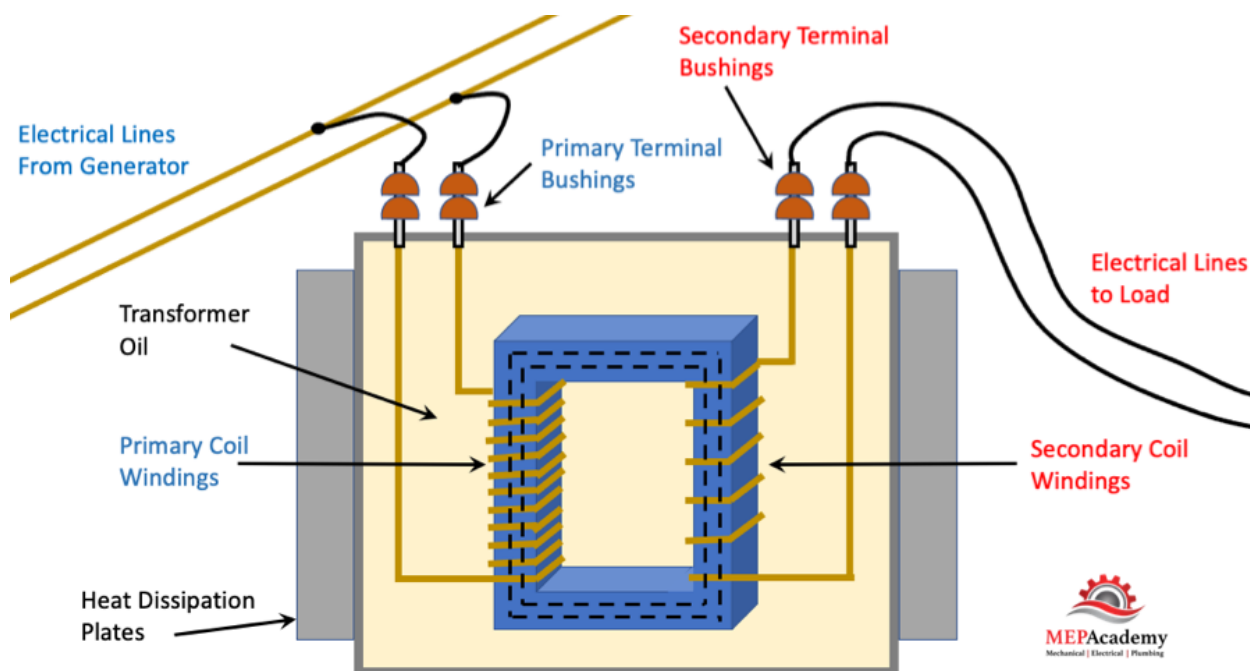
Transformatori u elektranama povećavaju napon za prijenos na velike udaljenosti, smanjujući gubitak energije, a u blizini točaka potrošnje smanjuju napon na sigurne razine. U industriji omogućuju posebne razine napona za strojeve i opremu te štite osjetljivu opremu izolacijom krugova.



Slika 2.2. Prikaz prijenosa električne energije.

Postoji nekoliko vrsta transformatora kao što su energetske transformatori koji se koriste u prijenosnim mrežama za primjenu pojačanog i sniženog napona, distributivni transformatori koji se koriste za distribuciju električne energije krajnjim korisnicima radeći na nižim naponima, mjerni transformator koji uključuju strujne i potencijalne transformatore koji se koriste za mjerenje i zaštitu te izolacijski transformatori koji omogućuju električnu izolaciju bez mijenjanja razine napona.

Transformatori su vrlo učinkoviti uređaji, često postižu učinkovitosti iznad 95%. Međutim, oni imaju gubitke, prvenstveno gubitke u jezgri. Redovito održavanje i testiranje osiguravaju pouzdanost i dugovječnost transformatora. Ključni testovi uključuju ispitivanje otpora izolacije, ispitivanje omjera zavoja transformatora i analizu ulja za one koji koriste hlađenje uljem. [1]



Slika 2.3. Uljem uronjen transformator.

2.3. Praćenje stanja transformatora

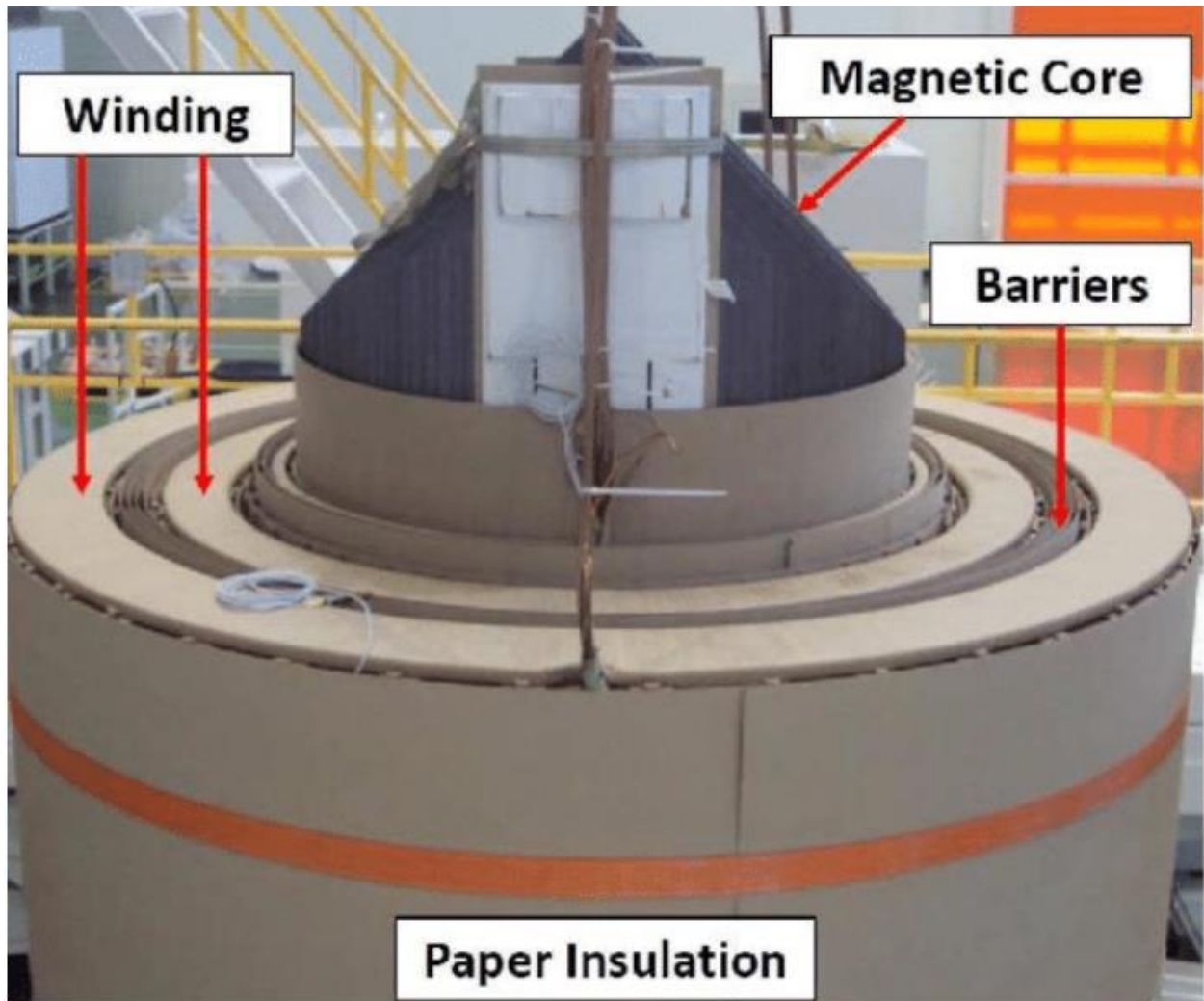
S obzirom na njihovu kritičnu ulogu, prevencija kvarova postaje prioritet za operatore elektroenergetskih sustava. Problem je što kvarovi transformatora mogu uzrokovati prekide u napajanju, financijske gubitke te sigurnosne rizike.

Izolacijski papir u transformatorima vremenom degradira zbog različitih toplinskih, električnih i kemijskih utjecaja. Kritičan indikator stanja papira je stupanj polimerizacije. Niski stupanj polimerizacije upućuje na smanjenu mehaničku čvrstoću papira. Stupanj polimerizacije se može mjeriti laboratorijskim metodama, ali su one često nepraktične za terensku upotrebu. Kao alternativa, kao indikatori degradacije mogu se koristiti koncentracije ugljikovih oksida (CO i CO₂), jer se oni oslobađaju tijekom razgradnje celuloze u papiru.



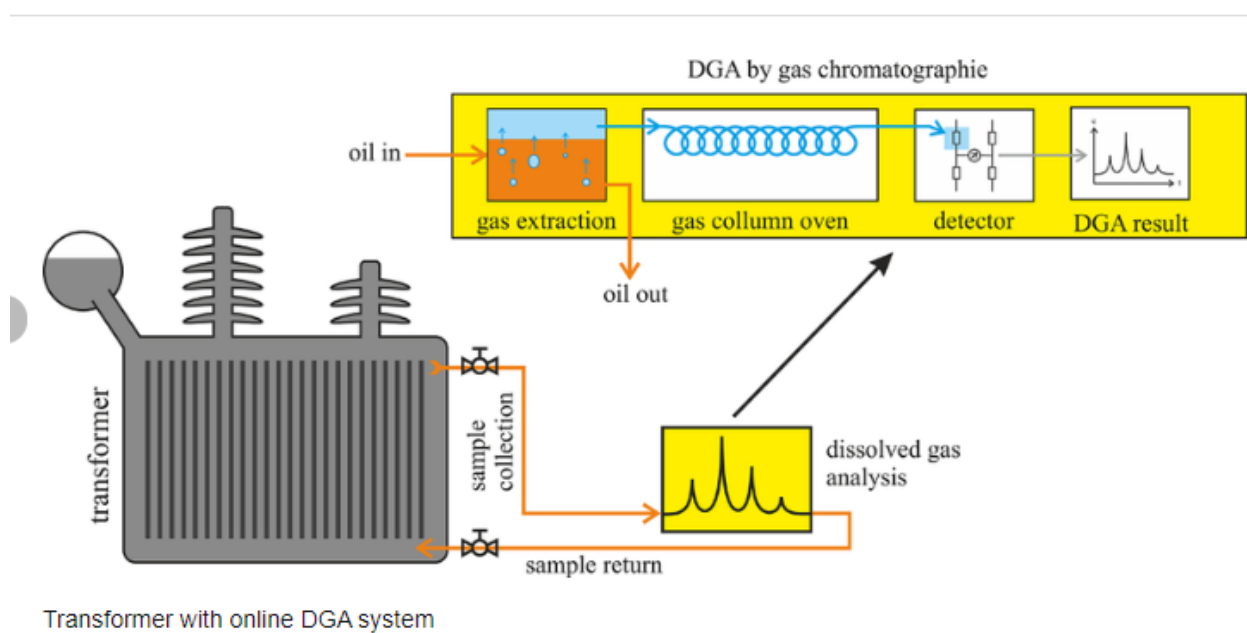
Slika 2.4. *Izolacijski papir.*

Još jedan važan pokazatelj degradacije izolacijskog papira su furani. Furani nastaju tijekom termalne razgradnje celuloze i njihova prisutnost u ulju transformatora može pružiti informacije o kvaliteti izolacije. Mjerenje koncentracija furana koristi se za procjenu stupnja razgradnje i potencijalne preostale životne dobi papira.



Slika 2.5. Primjena izolacijskog papira.

Analiza otopljenih plinova (DGA) je tehnika za rano otkrivanje kvarova transformatora koja uključuje mjerenje koncentracija različitih plinova, poput vodika, metana, acetilena, etilena i etana, koji se oslobađaju tijekom različitih kvarova. Standardna metoda za mjerenje ovih plinova je plinska kromatografija, ali radi se o metodi koja zahtijeva složenu opremu i specijalizirano osoblje pa se zbog toga kao alternativa koristi fotoakustička spektroskopija koja nudi brže i jednostavnije analize za terensku primjenu.

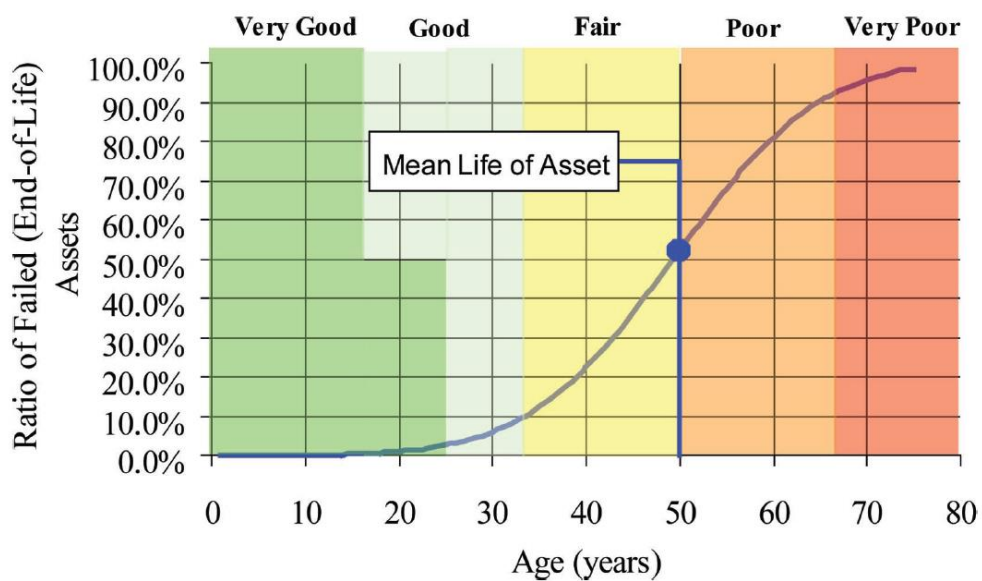


Slika 2.6. Prikaz DGA analize.

Analiza podataka otopljenih plinova koristi se za dijagnosticiranje specifičnih kvarova unutar samog transformatora. Key Gas metoda identificira ključne plinove povezane s određenim vrstama kvarova, dok Doernenburg metoda, Rogers metoda i Duvalov trokut pružaju detaljne algoritme za analizu omjera koncentracija različitih plinova i omogućuju preciznu dijagnostiku i identifikaciju specifičnih problema, što je bitno za pravovremeno održavanje i prevenciju kvarova.

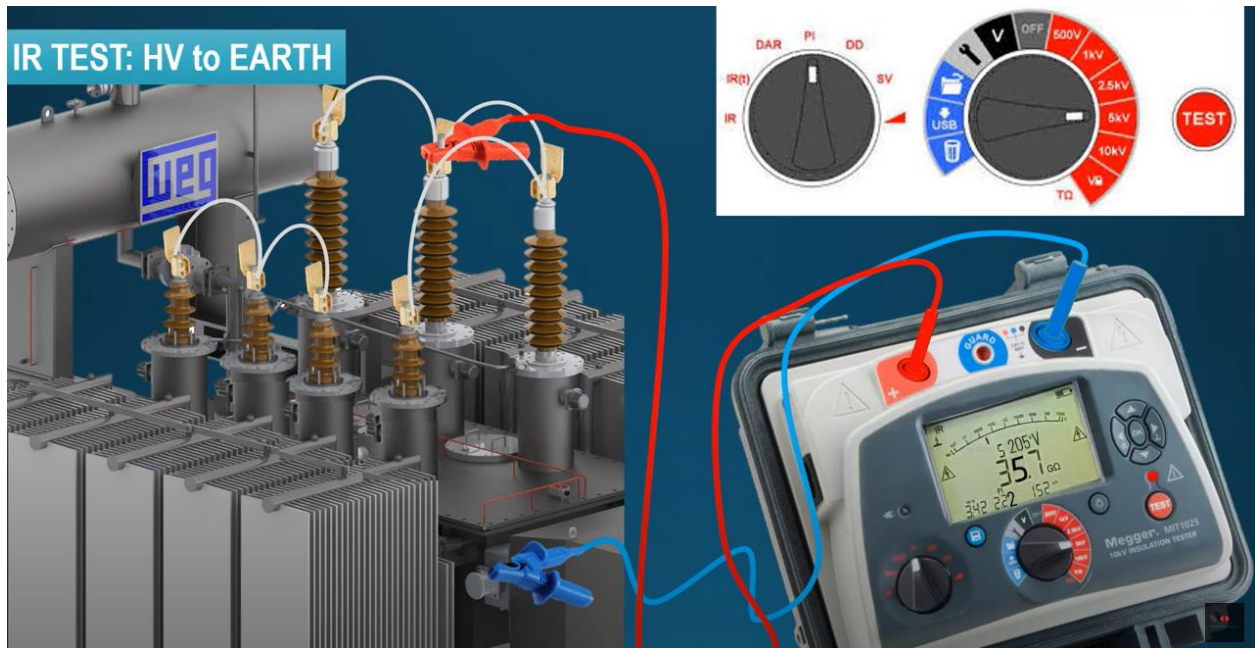
2.4. Indeks zdravlja transformatora

Sve više se tradicionalne metode nadzora i održavanja transformatora nadopunjuju i zamjenjuju naprednijim pristupima poput korištenja indeksa zdravlja, koji omogućuje objektivnu procjenu stanja transformatora. Indeks zdravlja ili Health Index se koristi za prikupljanje i analizu podataka dobivenih različitim metodama ispitivanja i pruža jasnu sliku o ukupnom zdravlju transformatora. Standardni vijek trajanja transformatora prema proizvođačima je između 25 i 40 godina, ali mnogi transformatori u pogonu prelaze te granice i dosežu 60 godina starosti. Bez obzira na to, stope kvarova ostaju niske što pokazuje da mnogi transformatori mogu funkcionirati i nakon očekivanog vijeka trajanja. Cilj je pružiti procjenu stanja koja uključuje ne samo tehničke podatke, nego i povijesne informacije o radu i održavanju transformatora. [2]



Slika 2.7. Odnos starosti i neuspjeha transformatora prema kategorijama.

Za izradu procjene stanja transformatora koriste se različite vrste ispitivanja i analiza. Mjerenje otpora izolacije (Megger ispitivanje) je metoda koja pomaže u procjeni stanja izolacijskog materijala transformatora dok vizualna inspekcija obuhvaća pregled spremnika, radijatora, ventilatora i drugih komponenti kako bi se otkrili vidljivi znakovi trošenja ili oštećenja.



Slika 2.8. Prikaz simulacije Megger ispitivanja izolacije transformatora.

2.5. Metode procjene i održavanja kvalitete ulja

Za potrebu ovog diplomskog rada najvažnije su metode procjene kvalitete ulja. Prema standardu IEC 60599 i IEEE Standardu C57.104, analiza otopljenih plinova (DGA) može identificirati probleme poput djelomičnog izboja, preopterećenja i pregrijavanja transformatora.

Rating Code	Condition	Description
A	Good	DAF < 1.2 [AU8: I would left align the last column on DGAF]
B	Acceptable	$1.2 \leq DGAF < 1.5$
C	Need Caution	$1.5 \leq DGAF < 2$
D	Poor	$2 \leq DGAF < 3$
E	Very poor	$DGAF \geq 3$

Slika 2.9. Utjecaj DGA faktora na ocjenu kvaliteta stanja transformatora.

Iako je DGA izuzetno korisna, sama po sebi nije dovoljna za potpuni uvid u stanje transformatora, pa se mora koristiti zajedno s drugim metodama ispitivanja. Analiza ulja obuhvaća mjerenje njegovih električnih, fizičkih i kemijskih svojstava prema standardima poput ASTM i IEC i služi za prepoznavanje potencijalnih problema prije nego što dođe do ozbiljnih kvarova, pružajući mogućnost preventivnog održavanja.

	$U \leq 69 \text{ kV}$	$69 \text{ kV} < U < 230 \text{ kV}$	$230 \text{ kV} \leq U$	Score (Si)
Dielectric Strength kV (2 mm gap)	≥ 45	≥ 52	≥ 60	1
	35–45	47–52	50–60	2
	30–35	35–47	40–50	3
	≤ 30	≤ 35	≤ 40	4
IFT dyne/cm	≥ 25	≥ 30	≥ 32	1
	20–25	23–30	25–32	2
	15–20	18–23	20–25	3
	≤ 15	≤ 18	≤ 20	4
Acid Number	≤ 0.05	≤ 0.04	≤ 0.03	1
	.05–0.1	0.04–1.0	0.03–.07	2
	0.1–0.2	1.0–0.15	0.07–.10	3
	≥ 0.2	≥ 0.15	≥ 0.10	4
Water content (ppm)	≤ 30	≤ 20	≤ 15	1
	30–35	20–25	15–20	2
	35–40	25–30	20–25	3
	≥ 40	≥ 30	≥ 25	4
Color	≤ 1.5			1
	1.5–2.0			2
	2.0–2.5			3
	≥ 2.5			4
Dissipation factor (%) 25 °C	≤ 0.1			1
	0.1–0.5			2
	0.5–1.0			3
	≥ 1.0			4

Slika 2.10. Metoda ocenjivanja temeljena na IEEE C57.106-2006..

Mjerenje furfurala u ulju koristi se za procjenu koncentracije furfurala koja ukazuje na degradaciju izolacije, što može biti znak da se transformator približava kraju svog trajanja. Također, mjerenje faktora disipacije ($\tan \delta$) važno je za praćenje električnih gubitaka u izolaciji transformatora. Ovo ispitivanje, zajedno s drugim mjerenjima kao što su otpor, temperatura i struja, pomaže u procjeni ukupnog stanja transformatora. [3]

Rating Code	Furaldehyde [ppm]	Age Years
A	0–0.1	Less than 20
B	0.1–0.25	20–40
C	0.25–0.5	40–60
D	0.5–1.0	More than 60
E	More than 1.0	—

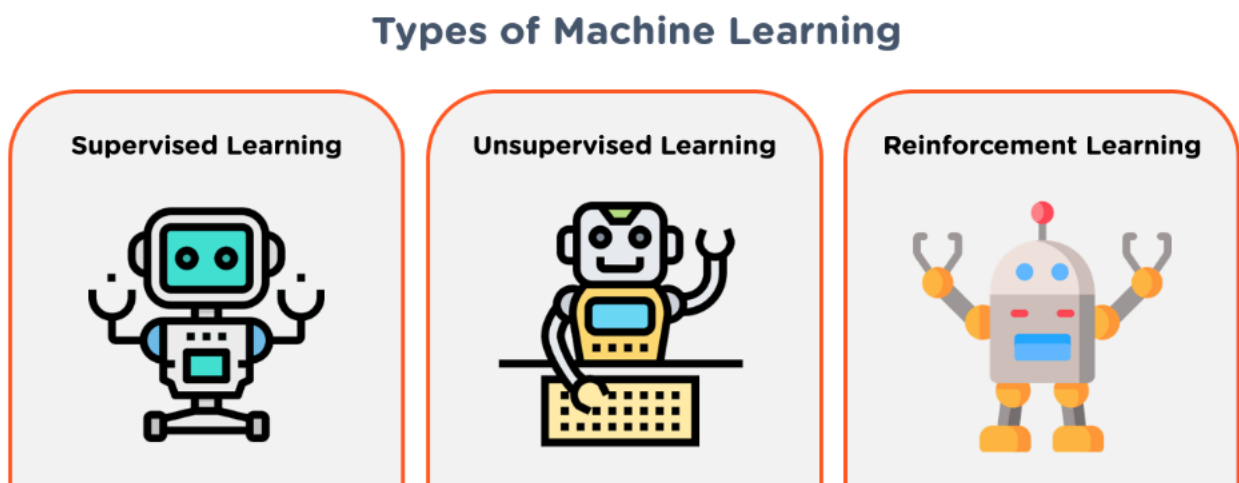
Slika 2.10. Metoda ccjenjivanja temeljena na razini furfurala.

3. RAZVOJ MODELA

Ovo poglavlje opisuje proces izrade modela strojnog učenja, od teorijskih osnova i različitih pristupa strojnog učenja. Opisuje se analiza i priprema podataka, a na kraju se uspoređuju različiti algoritami strojnog učenja, njihove performansame i kriterij za odabir najboljeg algoritma za izradu modela.

3.1. Strojno učenje

Strojno učenje jest programiranje računala tako da optimiziraju neki kriterij uspješnosti na temelju podatkovnih primjera ili prethodnog iskustva. U današnje vrijeme strojno ima različite primjene, te je jedno od najbrže rastućih polja računalne znanosti. Kada je riječ o podijeli, strojno učenje se dijeli na nadzirano, nenadzirano i podržano strojno učenje. [4]



Slika 3.1. Vrste strojnog učenja.

3.1.1. Nadzirano strojno učenje

Nadzirano učenje je temeljni koncept strojnog učenja koji omogućuje stvaranje modela sposobnog za donošenje odluka i predviđanje. Algoritmi uče na skupu podataka koji sadrže ulazne podatke zajedno s odgovarajućim izlaznim oznakama. Cilj ovog procesa je razviti model koji može precizno predviđati izlazne oznake za nove, njemu nepoznate ulazne podatke na temelju naučenog iz trening skupa podataka. Algoritam analizira podatke i stvara model koji može upariti ulazne podatke i izlazne oznake. Nakon treninga, model se testira na novim podacima kako bi se procijenila njegova točnost.

Kod nadziranog učenja razlikujemo klasifikaciju i regresiju. Klasifikacija se koristi kada su izlazne oznake kategoričke vrijednosti, kao što je prepoznavanje rukom pisanih brojeva ili kategorizacija emailova na spam i ne-spam. Dok se regresija primjenjuje kada su izlazne oznake kontinuirane vrijednosti, kao što je predviđanje cijena kuća ili automobila.

3.1.2. Nenadzirano strojno učenje

Nenadzirano učenje omogućuje istraživanje kompleksnih struktura i veza unutar skupa podataka, a svoju primjenu nalazi u analizi podataka i umjetnoj inteligenciji . Algoritmi analiziraju podatke bez unaprijed određenih izlaznih oznaka ili ciljeva. Cilj ove metode je otkriti skrivene obrasce, relacije i poveznice unutar skupa podataka. Za razliku od nadziranog učenja, gdje model trenira na temelju ulaznoizlaznih parova, nenadzirano učenje radi samo s ulaznim podacima. Algoritmi nenadziranog učenja pokušavaju grupirati podatke u kategorije ili pronaći pravila koja opisuju te podatke.

Ima široku primjenu u raznim područjima poput marketinga gdje se koristi za segmentaciju korisnika prema njihovom ponašanju, a u analizi tržišta koristi se za otkrivanje skrivenih obrazaca u potrošačkim podacima. Također nalazi svoju primjenu u kompresiji i vizualizaciji podataka te detekciji nepravilnosti.

3.1.3. Podržano strojno učenje

Podržano učenje je tip strojnog učenja koji se često primjenjuje u robotici i navigaciji. U podržanom učenju algoritam putem niza pokušaja i pogreška otkriva koji redoslijed akcija daje najbolje rezultate, a za cilj ima pronaći strategiju koja će dati najbolju dobit.

3.2. Analiza i podjela podataka

Podaci za strojno učenje su skup podataka koji se koriste za treniranje, testiranje i evaluaciju modela strojnog učenja. Podaci mogu biti strukturirani ili nestrukturirani. Strukturirani podaci su podaci koji se nalaze u tabličnom formatu i lako se mogu obrađivati pomoću matematičkih funkcija. Ovi podaci se najčešće koriste u strojnom učenju za klasifikaciju i regresiju dok su nestrukturirani podaci oni podaci koji nemaju jasnu strukturu i vrlo teško se obrađuju.

Radi što boljih rezultata i što preciznije točnosti modela korišten je kvalitetan skup podataka s preko 10 000 različitih uzoraka analiza ulja transformatora za potrebu izrade rada.

Zadatak je da na osnovu povijesnih podataka fizikalno-kemijskih analiza, koji su arhivirani u zadanom skupu, metodama strojnog učenja izraditi modele koji će ocijeniti stanje transformatora na osnovu izmjerenih parametara. Nasa ciljana vrijednost je kvaliteta ulja transformatora sto znaci da se radi o nadziranom strojnom učenju. Kako bi odredili radi li se o klasifikaciji ili regresiji moramo pogledati tip podatka ciljane vrijednosti. Pošto se radi o kategoričkim, a ne numeričkim vrijednostima možemo zaključiti da je potrebno izraditi klasifikacijski model.

Kako bi pripremili podatke za daljnju obradu bilo potrebno je napraviti manje izmjene. Uklonjen je stupac 'ID' jer nam je ID transformatora nepotreban za analizu kao i naručitelj te je napravljena pretvorba u odgovarajuće tipove podataka radi lakše obrade. Numeričke vrijednosti su iz string tipa podataka pretvorene u odgovarajuće vrijednosti. Stvaraju se dvije liste `categorical_columns` i `numerical_columns` koje se popunjavaju stvarnim vrijednostima podataka iz skupa podataka `X` koji odgovaraju imenima stupaca u odgovarajućim listama. Lista `categorical_columns` sadrži sve kategoričke varijable, dok `numerical_columns` sadrži sve numeričke varijable.

```
data = pd.read_csv(file_path)

categorical_columns = ['PROIZ', 'TR_BR', 'TR_TIP', 'BOJA', 'ZAKLJ']
numerical_columns = [
    'TR_SNAGA', 'TR_NAP2', 'TR_NAP3', 'TR_GOD', 'VODA', 'T_PAL', 'PB_1', 'PB_2',
    'PB_3', 'PB_4', 'PB_5', 'PB_6', 'ST_DEV', 'SR_VRIJ', 'POVRSINSKA_NAPETOST',
    'ZKL_ID', 'TRF_ID', 'TEMP_UZORKA'
]
```

Slika 3.2. Podijela podataka na kategoričke i numeričke.

Nadalje podaci su podijeljeni u kategorije temeljem tipa podataka u svakom stupcu. Varijabla `x` sadrži sve podatke iz skupa podataka, osim stupca koji je ciljna varijabla, dok je `y` ta ciljna varijabla.

```
X = data.drop(columns=['ZAKLJ'])
y = data['ZAKLJ']
```

Slika 3.3. Podijela odvajanje ciljane varijable.

Izvedba modela strojnog učenja ovisi o tome kako obrađujemo i unosimo različite vrste varijabli u model. Budući da većina modela strojnog učenja prihvaća samo numeričke varijable, pretprocesiranje kategoričkih varijabli je nužno. Kategoričke varijable moramo pretvoriti u brojeve tako da model može razumjeti i izdvojiti vrijedne informacije. Kako bi to postigli koristimo Encoder. Postoji nekoliko vrsta Encodera, a za potrebu ovog modela koristi se Label Encoder koji će svakoj kategoričkoj vrijednosti pridružiti odgovarajuću numeričku vrijednost.

```
label_encoders = {}
for col in categorical_columns:
    if col != 'ZAKLJ':
        le = LabelEncoder()
        data[col] = le.fit_transform(data[col])
        label_encoders[col] = le

target_encoder = LabelEncoder()
data['ZAKLJ'] = target_encoder.fit_transform(data['ZAKLJ'])
```

Slika 3.4. *Primjena Label Encodera na kategoričke vrijednosti.*

Kako bi algoritmi što bolje radili, odnosno kako bi model bio precizniji potrebno je skalirati podatke odnosno provesti standardizaciju podataka. Uporabom StandardScalera standardizirane su značajke skupa podataka tako da imaju srednju vrijednost 0 i standardnu devijaciju 1. Nakon što su srednja vrijednost i standardna devijacija izračunate, StandardScaler transformira podatke oduzimanjem srednje vrijednosti od svake značajke i dijeljenjem rezultata s njenom standardnom devijacijom.

```
scaler = StandardScaler()
X[numerical_columns] = scaler.fit_transform(X[numerical_columns])
```

Slika 3.5. *Primjena StandardScalera.*

Jedan od problema skupa podataka je i neuravnoteženost klasa. Kod modela klasifikacije često dolazi do situacije da su neke klase značajno manje zastupljeni od drugih. Kako bi se riješio ovaj problem koristi se Synthetic Minority Over-sampling Technique (SMOTE). osmišljen kako bi se povećala reprezentacija manjinskih klasa putem sintetičkih primjera, čime se poboljšava ravnoteža klasa. Za svaki manjinski primjer, SMOTE generira sintetičke primjere tako što nasumično odabire jednog od k najbližih susjeda.

```
smote = SMOTE(random_state=42, k_neighbors=2)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

Slika 3.6. *Primjena Synthetic Minority Over-sampling Technique.*

SMOTE se koristi s `k_neighbors=2`, što znači da će se sintetički primjeri generirati koristeći dva najbliža susjeda za svaki manjinski primjer.

Kod strojnog učenja zadani skup podataka je potrebno podijeliti na trening skup podataka na kojem će se model učiti i test skup podataka pomoću kojega se procjenjuje preciznost modela. Obično, oko 70% podataka se koristi za treniranje modela, a 30% za testiranje modela. Kod ovakvog načina podijele podataka s vremenom su uočeni nedostaci kod dobivenih modela. Ovisno o tome kako su podaci podijeljeni model može postati pristran, dok se procjena modela provodi samo jednom na odabranom testnom skupu i može dati krivu procjenu modela.

Kako bi se izbjegli navedeni problemi prilikom izrade modela koristenjena je Cross-validacija. Umjesto da jednostavne podijele podatke na trening i test skup, cross-validacija dijeli podatke na više podskupova i koristi te podskupove za višestruko treniranje i validaciju modela. To daje pouzdaniju procjenu performansi modela. Podaci se dijele na k jednakih dijelova i model se trenira k puta. Svaki put, jedan od podskupova se koristi kao validacijski skup, dok se preostalih $k-1$ koriste se za treniranje modela. Na taj način, svaki podskup je korišten točno jednom za validaciju i $k-1$ puta za treniranje. Na kraju da bi se dobila konačna procjena i točnost modela uzima se prosječna vrijednost svih testiranja što rezultira boljom procjenom performansi i nepristranosti modela. [5]

```
cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
scores = cross_val_score(model, X_resampled, y_resampled, cv=cv, scoring='accuracy')
model.fit(X_resampled, y_resampled)
```

Slika 3.7. *Primjena Cross-validacije.*

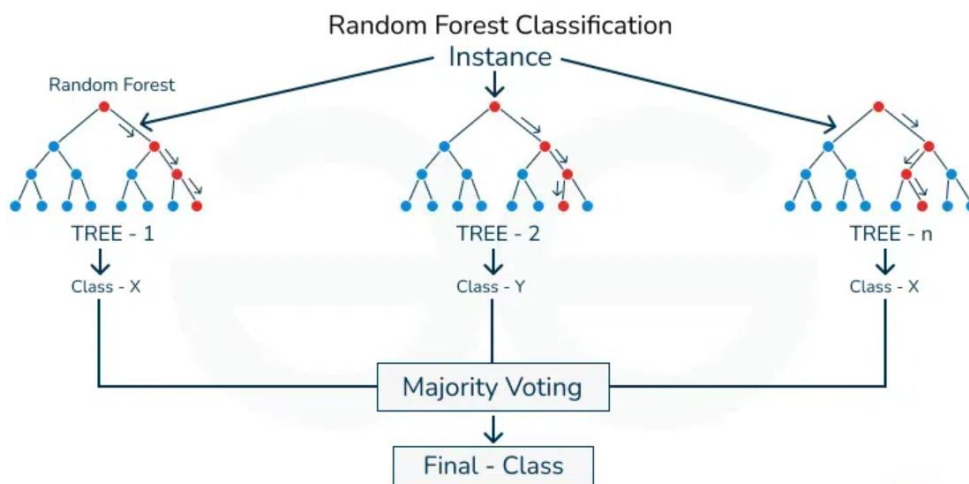
Podaci su podijeljeni u 3 podskupa, model se trenira 3 puta, svaki put s drugim podskupom kao validacijskim skupom.

3.3. Usporeba i odabir algoritma za izradu modela

U nadziranom strojnom učenju postoji mnogo algoritama koji se koriste pri treniranju modela, te nije lako nasumično odlučiti najbolji algoritam za odabrani model i potrebe modela zato što odabir ovisi i o skupu podataka s kojim se raspolaže. Kako bi se odabrao odgovarajući algoritam za model napravljena je usporedba preciznosti modela koristeći RandomForestClassifier, Support Vector Machine (SVM) i K-Nearest Neighbors (KNN).

3.3.1. RandomForestClassifier

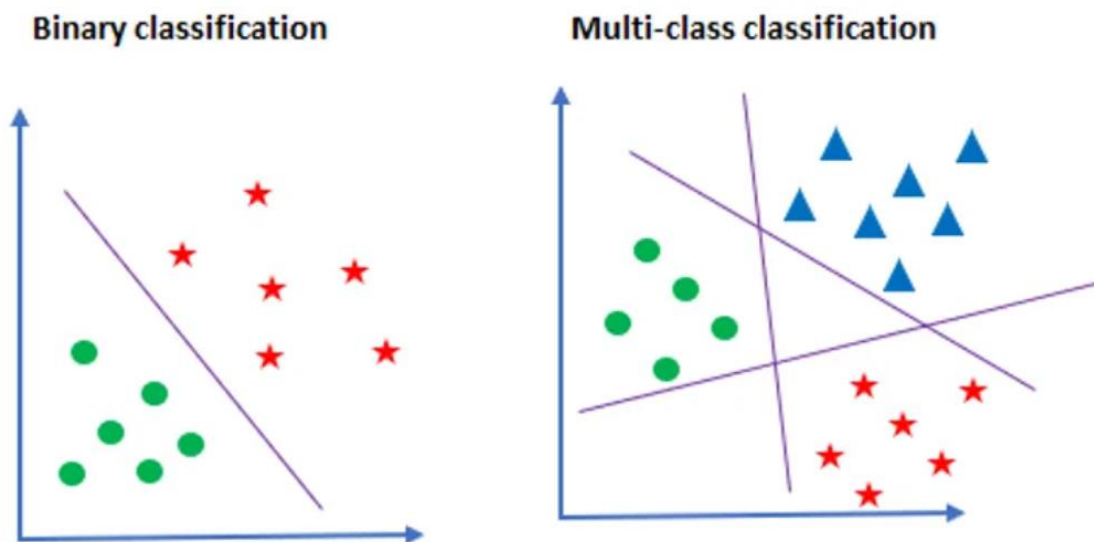
Random Forest je algoritam strojnog učenja koji se koristi za tipove zadataka klasifikacije. Predstavlja skup klasifikatora koji imaju strukturu stabla. Svako stablo u šumi daje svoj glas, dodjeljujući svakom ulazu najvjerojatnije oznake klase odnosno izlaza. Lako rukuje i numeričkim i kategoričkim podacima, a jedna od glavnih prednosti Random Foresta je ta što ne pati od prekomjernog uklapanja podataka zato što koristi slučajni odabir podskupova podataka za svako stablo. [6]



Slika 3.8. Princip rada RandomForestClassifier.

3.3.2. Support Vector Machine

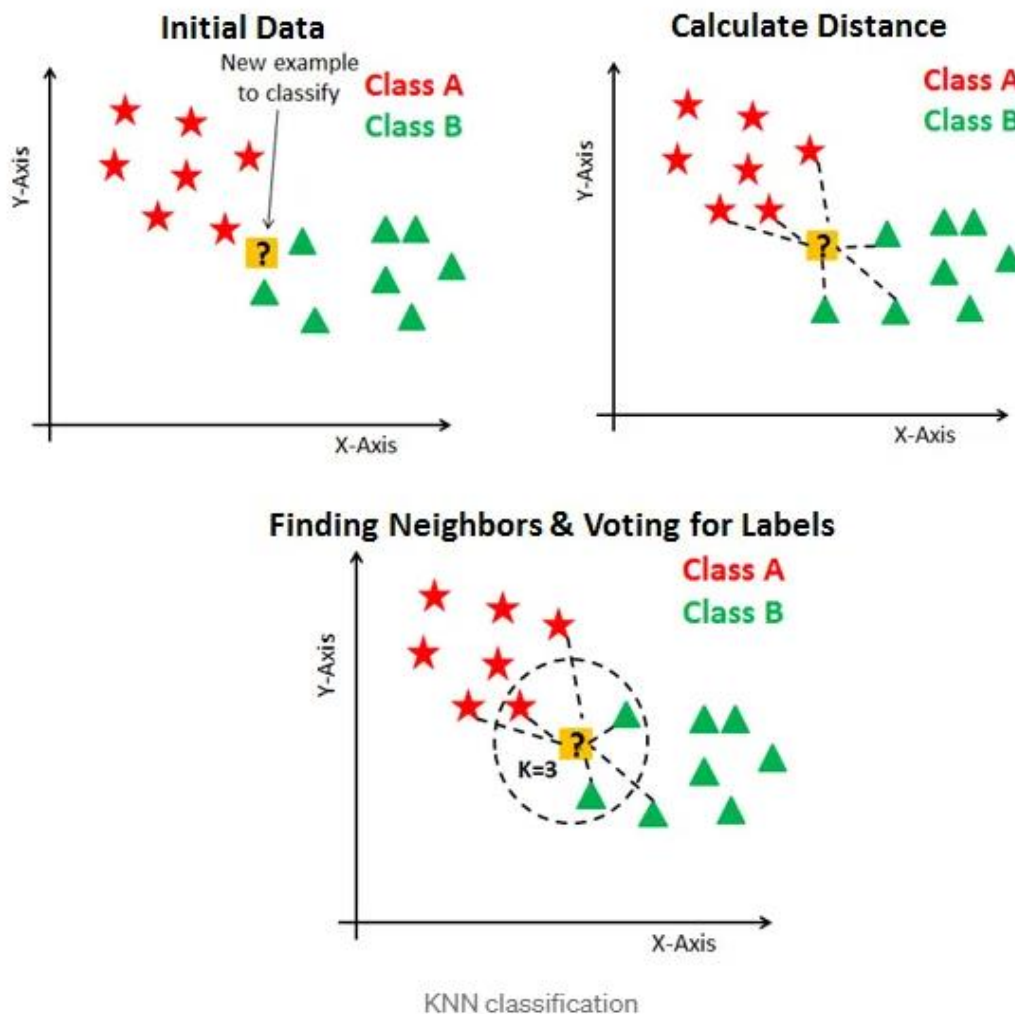
Support Vector Machine je algoritam strojnog učenja koji je dizajniran za razlikovanje između dvije klase. Međutim, SVM se može proširiti i koristiti za višeklasne probleme. One-vs-One strategija je tehnika koja omogućava korištenje Support Vector Machine algoritma za klasifikaciju kada imamo više od dvije klase na način da se provodi treniranje odvojenog SVM modela za svaki mogući par klasa u skupu podataka. Svaki binarni klasifikator se trenira koristeći samo primjere iz dva odabrana razreda, ignorirajući primjere iz drugih razreda. DOBRO / LOŠE će koristiti samo primjere iz klasa DOBRO i LOŠE, dok će ignorirati primjere iz klase ZADOVOLJAVA. Kada novi primjer treba biti klasificiran, svaki binarni klasifikator glasa za jednu od dvije klase koje promatra i konačna odluka je klasa s najviše glasova. [7]



Slika 3.9. Princip rada Support Vector Machine.

3.3.3. K-Nearest Neighbors

K-Nearest Neighbors klasifikator koristi se za klasifikaciju primjera dodjeljivanjem klase primjeru najbližih sličnih prethodno klasificiranih primjera. Koncept ovog algoritma se oslanja na parametar „k“ koji određuje koliko će susjeda biti izabrano za algoritam. Izbor parametra k ima velik utjecaj na učinkovitost algoritma. Veliki k smanjuje utjecaj varijance nastale slučajnim pogreškama, ali nosi rizik zanemarivanja malih obrazaca. Metoda za odabir odgovarajuće vrijednosti k je postizanje ravnoteže između prekomjernog i nedovoljnog uklapanja modela. Na primjer, različita ulja mogu se razlikovati po svojim svojstvima. Klasifikacija se temelji na sličnosti (euklidska udaljenost) između primjera i njegovih k najbližih susjeda, a konačna klasifikacija se određuje glasanjem većine susjeda. [8]



Slika 3.9. Princip K-Nearest Neighbors.

3.3.4. Rezultati usporedbe modela

```
models = {
    'RandomForest': RandomForestClassifier(random_state=42),
    'SVM': SVC(random_state=42),
    'KNN': KNeighborsClassifier(n_neighbors=5, weights='distance'),
}

cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)

results = {}

for model_name, model in models.items():
    scores = cross_val_score(model, X_resampled, y_resampled, cv=cv, scoring='accuracy')
    model.fit(X_resampled, y_resampled)
    results[model_name] = {
        'model': model,
        'accuracy': scores.mean(),
        'cv_scores': scores
    }
    print(f"Model: {model_name}")
    print(f"Cross-Validation Accuracy: {scores.mean()}")
    print(f"Cross-Validation Scores: {scores}")
    print('-' * 80)
```

Slika 3.10. Kod usporedbe modela.

```
Model: RandomForest
Cross-Validation Accuracy: 0.9728327982217831
Cross-Validation Scores: [0.97394418 0.97122746 0.97332675]
-----
Model: SVM
Cross-Validation Accuracy: 0.8051782333086358
Cross-Validation Scores: [0.80451963 0.80377871 0.80723635]
-----
Model: KNN
Cross-Validation Accuracy: 0.9113772948053017
Cross-Validation Scores: [0.91392937 0.91232403 0.90787849]
-----
```

Slika 3.11. Rezultati usporedbe.

Najbolje rezultate odnosno najveću preciznost postiže Random Forest model te će se isti koristiti i implementirati u aplikaciju.

4. IZRADA APLIKACIJE

U ovom poglavlju se prolazi kroz proces izrade aplikacije koja će pomoću odabranog Random Forest modela raditi predikciju kvalitete ulja prema unesenim parametrima od strane korisnika. Kod aplikacije je napisan u Python programskom jeziku uz uporabu Streamlit razvojnog okruženja.

4.1. Treniranje i pohrana modela

Kako bi se model mogao koristiti prvo je potrebno učitati i pripremiti skup podataka. Nakon treninga, model, Scaler i Enkodere je potrebno spremiti kako bi se mogli koristiti unutar aplikacije. Navedena pohrana odrađena je pomoću Picklea. Pickle je Python modul koji se koristi za pretvaranje Python objekta u niz bajtova, koji se mogu spremiti u datoteku. Također po potrebi moguće je pretvaranja tog niza bajtova natrag u Python objekt.

```
1 import pandas as pd
2 from sklearn.model_selection import StratifiedKFold, cross_val_score
3 from sklearn.preprocessing import LabelEncoder, StandardScaler
4 from sklearn.ensemble import RandomForestClassifier
5 from imblearn.over_sampling import SMOTE
6 import pickle
7
8 file_path = '/content/dataset.csv'
9
10 data = pd.read_csv(file_path)
11
12 categorical_columns = ['PROIZ', 'TR_BR', 'TR_TIP', 'BOJA', 'ZAKLJ']
13 numerical_columns = [
14     'TR_SNAGA', 'TR_NAP2', 'TR_NAP3', 'TR_GOD', 'VODA', 'T_PAL', 'PB_1', 'PB_2',
15     'PB_3', 'PB_4', 'PB_5', 'PB_6', 'ST_DEV', 'SR_VRIJ', 'POVRSINSKA_NAPETOST',
16     'ZKL_ID', 'TRF_ID', 'TEMP_UZORKA'
17 ]
18
19 label_encoders = {}
20 for col in categorical_columns:
21     if col != 'ZAKLJ':
22         le = LabelEncoder()
23         data[col] = le.fit_transform(data[col])
24         label_encoders[col] = le
25
26 target_encoder = LabelEncoder()
27 data['ZAKLJ'] = target_encoder.fit_transform(data['ZAKLJ'])
28
29 X = data.drop(columns=['ZAKLJ'])
30 y = data['ZAKLJ']
```

Slika 4.1. Prikaz koda za treniranje modela(1).

```

32 scaler = StandardScaler()
33 X[numerical_columns] = scaler.fit_transform(X[numerical_columns])
34
35
36 smote = SMOTE(random_state=42, k_neighbors=2)
37 X_resampled, y_resampled = smote.fit_resample(X, y)
38
39 model = RandomForestClassifier(random_state=42)
40
41 cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
42 scores = cross_val_score(model, X_resampled, y_resampled, cv=cv, scoring='accuracy')
43 model.fit(X_resampled, y_resampled)
44
45
46 with open('label_encoders.pkl', 'wb') as f:
47     pickle.dump(label_encoders, f)
48
49 with open('target_encoder.pkl', 'wb') as f:
50     pickle.dump(target_encoder, f)
51
52 with open('scaler.pkl', 'wb') as f:
53     pickle.dump(scaler, f)
54
55 with open('random_forest_model.pkl', 'wb') as f:
56     pickle.dump(model, f)
57
58 with open('feature_names.pkl', 'wb') as f:
59     pickle.dump(list(X.columns), f)

```

Slika 4.2. Prikaz koda za treniranje modela(2).

4.2. Streamlit aplikacija

Nakon pripreme samog modela i svih potrebnih elemenata, potrebno ih je implementirati u aplikaciju. Najbolji način za uporabu modela unutar aplikacije je korištenje Streamlita. Streamlit je Python razvojno okruženje za podatkovne znanstvenike, strojno učenje i za izradu podatkovnih aplikacija.

```
1 import streamlit as st
2 import pandas as pd
3 import pickle
4
5 with open('/Users/gauss/PycharmProjects/pythonProjecttest/main/label_encoders.pkl', 'rb') as f:
6     label_encoders = pickle.load(f)
7
8 with open('/Users/gauss/PycharmProjects/pythonProjecttest/main/target_encoder.pkl', 'rb') as f:
9     target_encoder = pickle.load(f)
10
11 with open('/Users/gauss/PycharmProjects/pythonProjecttest/main/scaler.pkl', 'rb') as f:
12     scaler = pickle.load(f)
13
14 with open('/Users/gauss/PycharmProjects/pythonProjecttest/main/random_forest_model.pkl', 'rb') as f:
15     model = pickle.load(f)
16
17 with open('/Users/gauss/PycharmProjects/pythonProjecttest/main/feature_names.pkl', 'rb') as f:
18     feature_names = pickle.load(f)
19
20 categorical_columns = ['PROIZ', 'TR_BR', 'TR_TIP', 'BOJA']
21 numerical_columns = [
22     'TR_SNAGA', 'TR_NAP2', 'TR_NAP3', 'TR_GOD', 'VODA', 'T_PAL', 'PB_1', 'PB_2',
23     'PB_3', 'PB_4', 'PB_5', 'PB_6', 'ST_DEV', 'SR_VRIJ', 'POVRSINSKA_NAPETOST',
24     'ZKL_ID', 'TRF_ID', 'TEMP_UZORKA'
25 ]
26
```

Slika 4.3. Prikaz koda za učitavanje modela, enkodera i skalera.

Prethodno spremljeni model te potrebne Enkodere i Scaler pomoću Pickle modula učitavamo u aplikaciju te navodimo kategoričke i numeričke varijable.


```

1 usage
27 def user_input_features():
28     input_data = {}
29     for feature in feature_names:
30         if feature in categorical_columns:
31             input_data[feature] = st.selectbox(f"{feature}", options=label_encoders[feature].classes_)
32         else:
33             input_data[feature] = st.number_input(f"{feature}", value=0.0)
34     return input_data
35

```

Slika 4.4. Prikaz funkcije `user_input_features()`.

Funkcija služi za prikupljanje ulaznih podataka od korisnika putem Streamlit aplikacije. Kreira se prazan dictionary koji će pohraniti korisničke unose. Korisniku se prikazuje selectbox za kategorizirane podatke i number_input za numeričke podatke pomoću kojih unosi potrebne parametre. Omogućava dinamičko prikupljanje korisničkih podataka, osigurava da su svi potrebni ulazi dostupni, te ih priprema za daljnju obradu i predikciju pomoću modela.

```

36  def main():
37      st.markdown(
38          """
39          <style>
40          .main {
41              background-color: #7FA0FF;
42          }
43          .stButton button {
44              background-color: #00838f;
45              color: white;
46              border: none;
47              padding: 10px 20px;
48              text-align: center;
49              text-decoration: none;
50              display: inline-block;
51              font-size: 16px;
52              margin: 4px 2px;
53              transition-duration: 0.4s;
54              cursor: pointer;
55          }
56          .stButton button:hover {
57              background-color: white;
58              color: black;
59              border: 2px solid #00838f;
60          }
61          </style>
62          """,
63          unsafe_allow_html=True
64      )
65
66      st.title("Procjena kvalitete ulja pomoću Random Forest modela")
67      st.write("Unesite potrebne parametre:")

```

Slika 4.5. Prikaz main funkcije(1).

```

71     if st.button('Procijeni'):
72         input_df = pd.DataFrame([input_data])
73         for col in categorical_columns:
74             input_df[col] = label_encoders[col].transform(input_df[col])
75
76         input_df[numerical_columns] = scaler.transform(input_df[numerical_columns])
77
78         input_df = input_df[feature_names]
79
80         prediction = model.predict(input_df)
81         prediction_label = target_encoder.inverse_transform(prediction)
82
83         hardcoded_prediction_label = prediction_label[0]
84         if hardcoded_prediction_label == 'DOBRO':
85             display_label = 'DOBRO'
86         elif hardcoded_prediction_label == 'ZADOVOLJAVA':
87             display_label = 'ZADOVOLJAVA'
88         elif hardcoded_prediction_label == 'LOSE':
89             display_label = 'LOŠE'
90
91         st.write(f"Predikcija: {display_label}")
92

```

Slika 4.6. Prikaz main funkcije(2).

Funkcija main() je glavni dio Streamlit aplikacije, koja definira cijeli tijek rada aplikacije, od prikaza korisničkog sučelja do obrade korisničkog unosa i prikaza rezultata. Unutar same funkcije definira se naslov aplikacije koji se prikazuje na vrhu te kratka uputa korisniku da unese potrebne parametre. Poziva se funkcija user_input_features() koja prikazuje kontrole za unos podatka od korisnika te se definira dugme na čiji pritisak se izvršava blok koda. Unutar tog bloka koda se nalazi pretvorba korisničkih unosa u DataFrame te se kategorizirani podaci transformiraju pomoću Enkodera te standardiziraju pomoću Scalera. Na kraju se osigurava da su podaci uneseni u pravilnom redu i vrši se predikcija koristeći Random Forest model i ispisuje se na sučelje.

```

93     if display_label == 'DOBRO':
94         st.write("""
95             Preporučeni postupci:
96             Nastavite s redovitim pregledima i analizom ulja prema predviđenom rasporedu. Redovito praćenje uključuje:
97             - Vizualne inspekcije
98             - Analiza otopljenih plinova (DGA)
99             - Mjerenje otpora izolacije
100            - Provjere fizikalnih i kemijskih svojstava ulja
101        """)
102        st.write("""Provodite standardne preventivne postupke održavanja transformatora, uključujući čišćenje i pregled vanjskih
103        komponenti te osigurajte da transformator radi u optimalnim uvjetima,
104        bez prekomjerne vlage, prašine i visokih temperatura koje mogu negativno utjecati na ulje.""")

```

Slika 4.7. Prikaz postupka održavanja za kategorizaciju DOBRO.

```

118     elif display_label == 'LOŠE':
119         st.write("""
120             Preporučeni postupci:
121             Odmah provedite detaljnu analizu uzroka degradacije ulja, uključujući:
122             - Detaljna analiza otopljenih plinova (DGA)
123             - Mjerenje faktora disipacije (tan δ) i kapacitivnosti
124             - Ostala relevantna ispitivanja izolacijskih svojstava
125             Poduzmite hitne korektivne mjere kao što su:
126             - Zamjena ulja: Uklanjanje starog i unošenje novog ulja koje ispunjava tehničke specifikacije.
127             - Regeneracija ulja: Provedite proces regeneracije kako bi se uklonili kontaminanti i obnovila svojstva ulja.
128        """)
129        st.write("""Detaljno pregledajte transformator zbog mogućih oštećenja koja su mogla uzrokovati degradaciju ulja i provedite
130        potrebne popravke.
131        Osigurajte optimalne radne uvjete za transformator kako bi se spriječilo ponavljanje problema. To uključuje kontrolu
132        temperature, vlage i drugih okolišnih čimbenika.""")
133

```

Slika 4.8. Prikaz postupka održavanja za kategorizaciju LOŠE.

```

185     elif display_label == 'ZADOVOLJAVA':
186         st.write("""
187             Preporučeni postupci:
188             Povećajte učestalost analiza ulja kako biste pratili promjene u njegovim svojstvima. To uključuje češću analizu
189             otopljenih plinova (DGA) i dodatna ispitivanja izolacijskih svojstava.
190             Provedite dodatne dijagnostičke testove kako biste identificirali potencijalne probleme. Ovi testovi uključuju:
191             - Detaljnija analiza otopljenih plinova (DGA)
192             - Mjerenje faktora disipacije (tan δ) i kapacitivnosti
193             - Ispitivanja fizikalnih i kemijskih svojstava ulja
194        """)
195        st.write("""Planirajte proaktivne mjere održavanja, kao što je filtracija ili regeneracija ulja, kako bi se poboljšala njegova kvaliteta.
196        Analizirajte povijest opterećenja transformatora i prilagodite radne uvjete ako je potrebno kako bi se smanjio stres na
197        izolacijski sustav.""")

```

Slika 4.9. Prikaz postupka održavanja za kategorizaciju ZADOVOLJAVA.

U ovisnosti o rezultatu predikcije modela, aplikacija osim rezultata predikcije ispisuje i preporučene postupke održavanja transformatora kako bi bio sto dugovječniji i kako bi poboljšali performanse transformatora.

Procjena kvalitete ulja pomoću Random Forest modela

Unesite potrebne parametre:

TR_SNAGA
52.00 - +

TR_NAP2
87.00 - +

TR_NAP3
44.00 - +

PROIZ
DP "ELEKTRA" ZAGREB ▾

TR_GOD
2023.00 - +

TR_BR
001 ▾

TR_TIP

Slika 4.10. Prikaz sučelja aplikacije.

DP "ELEKTRA" ZAGREB ▾

- ABB-TRAFO-BB
- AEG
- ARGO
- ASEA
- BERGAMO
- BRUSH TRANSFORMERS
- CSEPELI
- DINAMO
- MALO TAMNA

Slika 4.11. Prikaz odabira kategoričke varijable.

Procijeni

Predikcija: DOBRO

Preporučeni postupci: Nastavite s redovitim pregledima i analizom ulja prema predviđenom rasporedu. Redovito praćenje uključuje:

- Vizualne inspekcije
- Analiza otopljenih plinova (DGA)
- Mjerenje otpora izolacije
- Provjere fizikalnih i kemijskih svojstava ulja

Provodite standardne preventivne postupke održavanja transformatora, uključujući čišćenje i pregled vanjskih komponenti te osigurajte da transformator radi u optimalnim uvjetima, bez prekomjerne vlage, prašine i visokih temperatura koje mogu negativno utjecati na ulje.

Slika 4.12. Prikaz predikcije DOBRO.

Procijeni

Predikcija: ZADOVOLJAVA

Preporučeni postupci: Povećajte učestalost analiza ulja kako biste pratili promjene u njegovim svojstvima. To uključuje češću analizu otopljenih plinova (DGA) i dodatna ispitivanja izolacijskih svojstava. Provedite dodatne dijagnostičke testove kako biste identificirali potencijalne probleme. Ovi testovi uključuju:

- Detaljnija analiza otopljenih plinova (DGA)
- Mjerenje faktora disipacije ($\tan \delta$) i kapacitivnosti
- Ispitivanja fizikalnih i kemijskih svojstava ulja

Planirajte proaktivne mjere održavanja, kao što je filtracija ili regeneracija ulja, kako bi se poboljšala njegova kvaliteta. Analizirajte povijest opterećenja transformatora i prilagodite radne uvjete ako je potrebno kako bi se smanjio stres na izolacijski sustav.

Slika 4.13. Prikaz predikcije ZADOVOLJAVA.

Procijeni

Predikcija: LOŠE

Preporučeni postupci: Odmah provedite detaljnu analizu uzroka degradacije ulja, uključujući:

- Detaljna analiza otopljenih plinova (DGA)
- Mjerenje faktora disipacije ($\tan \delta$) i kapacitivnosti
- Ostala relevantna ispitivanja izolacijskih svojstava Poduzmite hitne korektivne mjere kao što su:
- Zamjena ulja: Uklanjanje starog i unošenje novog ulja koje ispunjava tehničke specifikacije.
- Regeneracija ulja: Provedite proces regeneracije kako bi se uklonili kontaminanti i obnovila svojstva ulja.

Detaljno pregledajte transformator zbog mogućih oštećenja koja su mogla uzrokovati degradaciju ulja i provedite potrebne popravke. Osigurajte optimalne radne uvjete za transformator kako bi se spriječilo ponavljanje problema. To uključuje kontrolu temperature, vlage i drugih okolišnih čimbenika.

Slika 4.13. Prikaz predikcije LOŠE.

ZAKLJUČAK

Cilj diplomskog rada je bio izraditi model na osnovu povijesnih podataka fizikalno-kemijskih analiza metodama strojnog učenja koji će ocijeniti stanje transformatora na osnovu izmjerenih parametara te izraditi odgovarajuću aplikaciju koja će omogućiti unos parametara, izračunati i ispisati odgovarajuću klasu ulja transformatora te uputiti u postupke održavanja s obzirom na zadanu kategorizaciju. Kod aplikacije napisan je u programskom jeziku Python, a u realizaciji korišteno je Streamlit razvojno okruženje. Nakon pripreme podataka, uspoređeno je nekoliko algoritama kako bi se odabrao odgovarajući algoritam za model koji će se koristiti u aplikaciji. Najbolje rezultate je pokazao Random Forest model koji ima preciznost od čak 97%. Aplikacija je jednostavna za korištenje te omogućava dinamičku analizu unesениh podataka te osim predikcije kvalitete ulja ispisuje i preporučene postupke održavanja transformatora. Strojno učenje predstavlja revoluciju u načinu na koji analiziramo podatke i donosimo odluke. Omogućava računalima da uče iz podataka, prepoznaju obrasce i donose predikcije s visokom preciznošću, neprestano evoluiraju i ima potencijal da značajno doprinese napretku znanosti i tehnologija.

U budućnosti aplikacija bi se mogla dodatno unaprijediti poboljšanjem kvalitete preciznosti modela iako je preciznost od 97% više nego dobra. Moguće je unaprijediti korisničko sučelje i vizualizaciju kako koji parametar utječe na rezultat predikcije. Koristeći Streamlit moguće je dodati više stranica te grafički prikazati podatke.

LITERATURA

- [1] Power Transformers Principles and Applications Autora John J. Winders, John Winders
- [2] A New Technique To Detect Loss Of Insulation Life In Power Transformers A. Bakar, Norazhar Bin
- [3] An approach to power transformer asset management using health index A. Jahromi; R. Piercy
- [4] Alpaydin, Ethem. Introduction to machine learning. MIT press, 2020.
- [5] Cross-validation1 Daniel Berrar Machine Learning Research Group School of Mathematics and Statistics The Open University, Milton Keynes.
- [6] NFORMATION PROCESSING IN AGRICULTURE 3 (2016) An improved random forest classifier for multi-class classification
- [7] Support Vector Machines for Multi-Class Pattern Recognition J. Weston and C. Watkins
- [8] Zhang Z. Introduction to machine learning: k-nearest neighbors. Ann Transl Med. 2016

SAŽETAK

Transformatori su jako bitne komponente elektroenergetskih sustava. Njihova pouzdanost i dugovječnost utječe na stabilnost i učinkovitost cijele elektroenergetske mreže. Stoga je bitno kontinuirano pratiti i procjenjivati stanje transformatora kako bi se osigurala pravovremena intervencija i spriječili kvarovi koji mogu uzrokovati ozbiljne poremećaje i gubitke. Nakon upoznavanja s pojmovima strojnog učenja i obrade podataka naredni cilj je bio metodama strojnog učenja razviti model i aplikaciju koja će predvidjeti stanje transformatora na temelju analiza ulja te predložiti i uputiti u potrebne postupke održavanja. Uspoređeno je nekoliko algoritama za klasifikaciju, a najbolje rezultate je imao Random Forest algoritam koji je potom korišten u izradi aplikacije. Aplikacije je jednostavna za korištenje te upućuje korisnika kako bi trebao postupiti. Aplikacija je realizirana u Streamlit razvojnom okruženju koristeći Python programski jezik.

Ključne riječi: transformator, ulje, strojno učenje, klasifikacija, Python, Streamlit, održavanje

ABSTRACT

Prediction of maintenance procedures for oil-immersed transformers based on physical-chemical analyses

Transformers are very important components of power systems. Their reliability and longevity affect the stability and efficiency of the entire power network. Therefore, it is essential to continuously monitor and assess the condition of transformers to ensure timely intervention and prevent failures that could cause serious disruptions and losses. After familiarizing ourselves with the concepts of machine learning and data processing, the next goal was to develop a model and application using machine learning methods to predict the condition of transformers based on oil analysis and to suggest and guide the necessary maintenance procedures. Several classification algorithms were compared, and the Random Forest algorithm produced the best results, which was then used in the application development. The application is easy to use and guides the user on how to proceed. The application was implemented in the Streamlit development environment using the Python programming language.

Key words: transformer, oil, machine learning, classification, Python, Streamlit, maintenance