

# Razvoj korisničkog sučelja prediktivnog održavanja za primjenu konvolucijske neuronske mreže

---

Lušetić, Tajana

Master's thesis / Diplomski rad

2020

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:580962>

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

*Download date / Datum preuzimanja:* **2024-04-20**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Tajana Lušetić**

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

Mentori:

Prof. dr. sc. Dragutin Lisjak

Student:

Tajana Lušetić

Zagreb, 2020.

Izjavljujem da sam ovaj rad izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru prof. dr. sc. Dragutinu Lisjaku i asistentu dr.sc. Davoru Kolaru na pruženoj pomoći i ustupljenim informacijama prilikom pisanja rada.

Također, zahvaljujem se dečku Nikoli, prijateljici Anastaziji i cijeloj obitelji na strpljenju i pruženoj potpori tijekom studiranja.

Tajana Lušetić



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

|                                                              |         |
|--------------------------------------------------------------|---------|
| Sveučilište u Zagrebu<br>Fakultet strojarstva i brodogradnje |         |
| Datum:                                                       | Prilog: |
| Klasa:                                                       |         |
| Ur. broj:                                                    |         |

## DIPLOMSKI ZADATAK

Student: **TAJANA LUŠETIĆ** Mat. br.: **0035199869**

Naslov rada na hrvatskom jeziku: **Razvoj korisničkog sučelja prediktivnog održavanja za primjenu konvolucijske neuronske mreže**

Naslov rada na engleskom jeziku: **Development of predictive maintenance user interface for application of convolutional neural network**

Opis zadatka:

Prediktivno održavanje po stanju novija je strategija održavanja temeljena na mjerenju i analizi parametara s ciljem planiranja aktivnosti održavanja na način da se optimizira životni vijek opreme. Razvojem tehnologija za prikupljanje velike količine podataka, te njihovo tumačenje primjenom algoritama iz područja umjetne inteligencije, potaknuo je trend razvoja prediktivnog održavanja stanja opreme usmjerenog prema tzv. "inteligentnoj predikciji". Današnji računalni sustavi sa sposobnošću predviđanja stanja opreme (engl. Prognostics and Health Management, PHM) u stanju su samostalno prepoznati anomaliju u radu opreme te predložiti određene aktivnosti održavanja kako bi se sve funkcionalnosti opreme održale na zadovoljavajućoj razini. Uočeni nedostatak u razvoju tehnologije je da trenutno nema jedinstvenog korisničkog sučelja koji bi objedinjavao module za prikupljanje i tumačenje podataka te jednostavnu integraciju pojedinih algoritama za strojno učenje. U skladu s navedenim, u radu je potrebno:

1. Opisati PHM sustave.
2. Objasniti strukturu potrebne ulazne datoteke za prikupljanje podataka.
3. Opisati algoritam konvolucijske neuronske mreže (CNN).
4. Primjenom razvojne okoline Matlab, razviti grafičko korisničko za CNN-algoritam koje treba imati sljedeće funkcionalnosti:
  - a) Učitavanje podataka iz dostupne ulazne datoteke,
  - b) definicija hiperparametara za učenje,
  - c) integracija algoritma za učenje,
  - d) evaluacija, izračun i prikaz performansi naučenog modela,
  - e) učitavanje i primjena naučenog modela na novom ulaznom skupu podataka.
5. Na temelju rezultata razvoja modela dati zaključak.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:  
14. studenog 2019.

Rok predaje rada:  
16. siječnja 2020.

Predviđeni datum obrane:  
20. siječnja do 24. siječnja 2020.

Zadatak zadao:  
prof. dr. sc. Dragutin Lisjak

Predsjednica Povjerenstva:  
prof. dr. sc. Biserka Runje

## SADRŽAJ

|                                                                               |      |
|-------------------------------------------------------------------------------|------|
| SADRŽAJ .....                                                                 | I    |
| POPIS SLIKA .....                                                             | III  |
| POPIS KRATICA .....                                                           | VI   |
| SAŽETAK .....                                                                 | VII  |
| SUMMARY .....                                                                 | VIII |
| 1. UVOD .....                                                                 | 1    |
| 2. PREDIKTIVNO ODRŽAVANJE PO STANJU .....                                     | 3    |
| 3. RAČUNALNI SUSTAV SA SPOSOBNOSTU PREDVIĐANJA STANJA OPREME ....             | 5    |
| 3.1. Senzori .....                                                            | 8    |
| 3.1.1. Mjerenje vibracija .....                                               | 9    |
| 3.1.2. Kontrola temperature .....                                             | 10   |
| 3.1.3. Kontrola podmazivanjem .....                                           | 10   |
| 3.2. Analiza računalnog sustava sa sposobnošću predviđanja stanja opreme..... | 11   |
| 3.2.1. Predobrada podataka i izdvajanje značajki .....                        | 11   |
| 3.2.2. Procjena stanja opreme.....                                            | 13   |
| 3.2.3. Dijagnostika .....                                                     | 14   |
| 3.2.4. Prognoziranje .....                                                    | 14   |
| 4. PRAĆENJE I DIJAGNOSTIKA KVAROVA .....                                      | 16   |
| 4.1. Dijagnostika kvarova temeljena na fizičkim modelima.....                 | 17   |
| 4.2. Dijagnostika kvarova temeljena na podacima.....                          | 18   |
| Slika 11. Tehnologije dijagnostike kvarova temeljene na podacima [3] .....    | 19   |
| 4.2.1. Statistički pristup .....                                              | 19   |
| 4.2.2. Strojno učenje bez nadzora.....                                        | 21   |
| 4.2.3. Strojno učenje pod nadzorom.....                                       | 24   |
| 5. KONVOLUCIJSKA NEURONSKA MREŽA .....                                        | 31   |
| 5.1. Konvolucijski sloj .....                                                 | 32   |
| 5.2. Sloj sažimanja.....                                                      | 33   |
| 5.3. Izlazni sloj .....                                                       | 34   |
| 5.4. Određivanje hiperparametara modela .....                                 | 35   |
| 6. RAZVOJ KORISNIČKOG SUČELJA .....                                           | 38   |
| 6.1. Razvojna okolina - <i>MATLAB Application Designer</i> .....              | 39   |
| 6.2. Struktura i učitavanje ulaznih podataka .....                            | 39   |
| 6.3. Podjela podataka za učenje, testiranje i validaciju .....                | 42   |
| 6.4. Određivanje hiperparametara za učenje mreže.....                         | 43   |
| 6.5. Integracija algoritma za učenje.....                                     | 44   |
| 6.5.1. Konvolucijski blokovi.....                                             | 46   |
| 6.5.2. Slojevi sažimanja .....                                                | 46   |
| 6.5.3. Potpuno povezani sloj i SoftMax.....                                   | 46   |
| 6.6. Evaluacija, izračun i prikaz performansi novog modela.....               | 46   |
| 6.7. Učitavanje i primjena naučenog modela .....                              | 48   |

---

|                                    |    |
|------------------------------------|----|
| 6.8. Kompajliranje .....           | 49 |
| 6.9. Primjer rada aplikacije ..... | 51 |
| 7. ZAKLJUČAK.....                  | 60 |
| LITERATURA.....                    | 62 |
| PRILOZI .....                      | 64 |

**POPIS SLIKA**

|           |                                                                                                |    |
|-----------|------------------------------------------------------------------------------------------------|----|
| Slika 1.  | Opći slučaj određivanja i značenja granica mjernog parametra [2] .....                         | 3  |
| Slika 2.  | Postupak uvođenja održavanja po stanju kontrolom parametra [2] .....                           | 4  |
| Slika 3.  | Primjer analize stabla kvarova [7] .....                                                       | 6  |
| Slika 4.  | Model obrade podataka prema ISO 13374-1 [3] .....                                              | 8  |
| Slika 5.  | Procjena vibracijskog stanja sustava prema ISO 10816 [4] .....                                 | 9  |
| Slika 6.  | Primjer sustava za mjerenje vibracija ležajeva SPM – Leonova i Brüel & Kjær [2] .....          | 10 |
| Slika 7.  | Zadaci u analizi PHM sustava [8] .....                                                         | 11 |
| Slika 8.  | Opća struktura otkrivanja i identifikacije kvarova [3] .....                                   | 16 |
| Slika 9.  | Klasifikacija sustava za predviđanje stanja opreme [11] .....                                  | 17 |
| Slika 10. | Tehnologije dijagnostike kvarova temeljene na fizičkom modelu [3] .....                        | 18 |
| Slika 11. | Tehnologije dijagnostike kvarova temeljene na podacima [3] .....                               | 19 |
| Slika 12. | Mapiranje dvodimenzionalnih podataka u jednodimenzionalne pomoću PCA [11] .....                | 22 |
| Slika 13. | Primjer strukture SOM-a [13] .....                                                             | 23 |
| Slika 14. | Primjer t-SNA [14] .....                                                                       | 24 |
| Slika 15. | Neuronske mreže za problem gustoće vjerojatnosti temeljen na Gaussovoj distribuciji [11] ..... | 25 |
| Slika 16. | Primjer neuronske mreže za problem klasifikacije [11] .....                                    | 26 |
| Slika 17. | Klasifikacije dvije klase korištenjem SVM-a [16] .....                                         | 26 |
| Slika 18. | Princip dijagnostike korištenjem SVM-a [16] .....                                              | 27 |
| Slika 19. | Grafički model za regresiju Gaussovog procesa [17] .....                                       | 28 |
| Slika 20. | Primjer Bayesove mreže [18] .....                                                              | 28 |
| Slika 21. | Modeliranje metodologije dijagnostike kvarova na temelju Bayesove mreže [18] .....             | 29 |
| Slika 22. | Opća struktura konvolucijskih neuronskih mreža [21] .....                                      | 31 |
| Slika 23. | Proces konvolucije i nastajanje mape značajki [22] .....                                       | 32 |
| Slika 24. | ReLU aktivacijska funkcija .....                                                               | 33 |
| Slika 25. | Obrada slike kroz konvoluciju, ReLU i sažimanje [22] .....                                     | 34 |
| Slika 26. | Softmax funkcija .....                                                                         | 35 |
| Slika 27. | Glavni zadaci razvijenog korisničkog sučelja .....                                             | 38 |
| Slika 28. | Prikaz prikupljenih podataka koji su spremljeni u CVS datoteku .....                           | 41 |
| Slika 29. | Primjer pozadinskog koda za pretvaranje CSV datoteke u MAT format .....                        | 42 |
| Slika 30. | Mape s 220 MAT datoteka .....                                                                  | 42 |
| Slika 31. | Proces treniranja neuronske mreže .....                                                        | 45 |
| Slika 32. | Blok shema pripreme za učenje optimizacijom parametra [21] .....                               | 45 |
| Slika 33. | Matrica zabune .....                                                                           | 47 |
| Slika 34. | Primjena modela i prikaz grafa .....                                                           | 49 |
| Slika 35. | Glavno sučelje .....                                                                           | 50 |
| Slika 36. | Dijagrami procesa za početno i kontinuirano korištenje korisničkog sučelja .....               | 51 |
| Slika 37. | Dio korisničkog sučelja za pripremu podataka .....                                             | 52 |
| Slika 38. | Učitavanje i podjela podataka za učenje mreže .....                                            | 53 |
| Slika 39. | Dio sučelja za određivanja hiperparametara, učenje, evaluaciju i spremanje mreže .....         | 54 |
| Slika 40. | Evaluacija modela .....                                                                        | 54 |
| Slika 41. | Analiza mreže .....                                                                            | 55 |
| Slika 42. | Učitavanja novih podataka i klasifikacija .....                                                | 56 |
| Slika 43. | Primjer sadržaja spremljene datoteke .....                                                     | 56 |



---

|           |                                                          |    |
|-----------|----------------------------------------------------------|----|
| Slika 44. | Grafički prikaz procenjenih klasa za pojedine dane ..... | 57 |
|-----------|----------------------------------------------------------|----|

---

POPIS TABLICA

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| Tablica 1. Metode za predobradu podataka [8].....                            | 11 |
| Tablica 2. Metode za izdvajanje značajki [8] .....                           | 12 |
| Tablica 3. Algoritmi za procjenu stanja opreme [8].....                      | 13 |
| Tablica 4. Rasponi hiperparametra [21] .....                                 | 43 |
| Tablica 5. Vrijednosti hiperparametra [21].....                              | 43 |
| Tablica 6. Sadržaj spremljenih datoteka koje služe za praćenje sustava.....  | 57 |
| Tablica 7. Mogući problemi i rješenja pri upotrebi korisničkog sučelja ..... | 58 |

---

**POPIS KRATICA**

| <b>Kratika</b> | <b>Opis</b>                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------|
| ALT            | Ispitivanje ubrzanog života (eng. Accelerated life test)                                          |
| CBM            | Održavanje po stanju (eng. Condition-based maintenance)                                           |
| CSV            | Vrijednosti odvojene zarezom (eng. Comma seperated value)                                         |
| FMEA           | Analiza uzroka i posljedica kvarova (eng. Failure mode and effects analysis)                      |
| FTA            | Stablo kvarova (eng. Fault tree analysis)                                                         |
| HiL            | Hardware in the loop                                                                              |
| HMM            | Skriveni Markov model (eng. Hidden Markov model)                                                  |
| IoT            | Internet stvari (eng. Internet of things)                                                         |
| PCA            | Analiza glavnih sastavnica (eng. Principal component analysis)                                    |
| PHM            | Računalni sustav sa sposobnošću predviđanja stanja opreme (eng. Prognostic and health management) |
| RTD            | Detektori otpora temperature (eng. Resistive temperature detectors)                               |
| RUL            | Preostali korisni život (eng. Remaning useful life)                                               |
| SOM            | Samoorganizirajuće karte (eng. Self-organizing maps)                                              |
| SPE            | Prediktivna kvadratna pogreška (eng. Squared prediction error)                                    |
| SVD            | Podaci potpornih vektora (eng. Support vector dana)                                               |
| SVM            | Metoda potpornih vektora (eng. Support vector machine)                                            |

---

**SAŽETAK**

Tema ovog diplomskog rada je razvoj grafičkog korisničkog sučelja za prediktivno održavanje. Prediktivno održavanje se sastoji od praćenja stvarnog mehaničkog stanja, radne učinkovitosti i drugih pokazatelja radnog stanja strojeva. Kako bi se mogli dijagnosticirati kvarovi strojeva, razvijene su razne metode od kojih su trenutačno najzanimljivije one koje su temeljene na podacima. Jedna od takvih metoda je konvolucijska neuronska mreža koja koristi matematičku operaciju konvoluciju, sažimanje i potpuno povezivanje slojeva za pretvaranje ulaznih slika ili signala sa senzora za dobivanje izlaza. Razvijeno korisničko sučelje omogućuje nadzor stanja opreme uz pomoć razvijenog i u grafičko korisničko sučelje implementiranog algoritma konvolucijske neuronske mreže. Korisničko sučelje omogućuje pripremu ulaznih podataka koje računalna okolina može koristiti, učenje neuronske mreže pomoću pripremljenih podataka, evaluacija novog modela te primjenu modela na novim podacima.

Ključne riječi: *Prediktivno održavanje, održavanje po stanju, predviđanja stanja opreme, strojno učenje, konvolucijske neuronske mreže, korisničko sučelje, MATLAB*

---

**SUMMARY**

The topic of this master thesis is the development of graphical user interface for predictive maintenance. Predictive maintenance consists of monitoring the actual mechanical condition, operating efficiency, and other indicators of the working condition of the machines. In order to be able to diagnose machine failures, various methods have been developed, of which currently the most interesting are those based on data. One such method is a convolutional neural network that uses mathematical operation, convolution, pooling, and softmax to convert input images or signals from sensors to obtain output. The developed application enables monitoring the condition of the equipment with the help of the convolution neural network algorithm developed in the graphical user interface. The application enables the preparation of input data that the program can use, learning neural networks using the prepared data, evaluating a new model and applying the model to new data.

*Key words: Predictive maintenance, condition-based maintenance prognostics and health management, machine learning, convolutional neural networks, user interface, MATLAB*

## 1. UVOD

Odlika 21. stoljeća je razvoj tehnologija i metoda za smanjenje troškova održavanja kako bi se povećala sigurnost i pouzdanost različite imovine odnosno proizvoda. Kako bi se to osiguralo, razvijaju se područja poput prediktivnog održavanja i sustava za nadzor i predviđanje stanja opreme. Kvarovi mogu uzrokovati velike štete sustavima, okolini i korisnicima zbog čega će predviđanje istih biti neizostavan dio svakog poslovanja. Sustavi za predviđanje već su postali popularni u različitim industrijama poput zrakoplovstva, proizvodnje, prometa i energetici. Bez obzira na primjenu, zajednička značajka takvih sustava je pretvaranje sirovih podataka u korisne informacije koje olakšavaju donošenje odluka. Takva primjena moguća je prikupljanjem velike količine podataka pomoću senzora, filtriranjem i analiziranjem podataka te donošenje odluka o budućim radnjama održavanja.

Drugo poglavlje služi kao uvod u temu prediktivnog održavanja po stanju, a u trećem poglavlju bit će opisan sustav sa sposobnošću predviđanja stanja opreme čije aktivnosti omogućuju i podržavaju održavanje po stanju. Glavne aktivnosti takvoga sustava su predobrada prikupljenih podataka sa senzora, procjena stanja opreme, dijagnostika i prognoziranje.

U četvrtom poglavlju će se detaljno opisati dijagnostike kvarova koja se može temeljiti na fizičkom modelu ili podacima. Postoje mnoge tradicionalne statističke metode na temelju kojih se obrađuju podaci, ali s obzirom na zahtjeve tržišta, koji su nametnuti brzinom i količinom prikupljenih podataka, potrebno je razvijati i primjenjivati nove metode. Jedna od takvih metoda je strojno učenje koje omogućava iskorištavanje korisnih informacija za predviđanje budućih događaja.

Daljnji razvoj strojnog učenja donosi pojmove kao što su duboko strojno učenje i konvolucijske neuronske mreže, koje se mogu koristiti za obradu slika ili senzorskih podataka u obliku matrice te će one biti objašnjene u petom poglavlju.

Uočeni nedostatak u razvoju tehnologija je da trenutno nema jedinstvenog korisničkog sučelja koji bi objedinio module za prikupljanje i tumačenje podataka te jednostavnu integraciju pojedinih algoritama strojnog učenja.

U ovom radu koristit će se algoritam konvolucijske neuronske mreže koji će biti integriran u korisničko sučelje razvijeno u razvojnoj okolini MATLAB. Detaljna izrada korisničkog sučelja opisan će se u šestom poglavlju. U korisničkom sučelju će biti moguće pripremiti podatke za učenje neuronske mreže, učiti neuronsku mrežu, napraviti evaluaciju mreže i na samom kraju primijeniti neuronske mreže na novim podacima. Očekuje se kako kontinuiranom primjenom

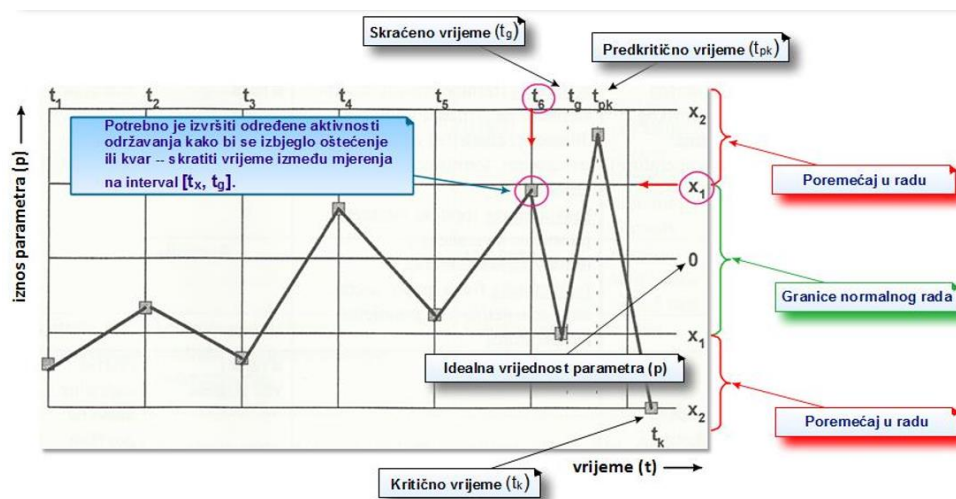
razvijenog grafičkog korisničkog sučelja odnosno naučenih modela na novim podacima sa senzora može unaprijediti donošenje odluka u održavanju određenog sustava koji se promatra. Takav razvijeni sustav će predstavljati korak prema razvoju sveobuhvatnog računalnog sustava sa sposobnošću predviđanja stanja održavane opreme.

## 2. PREDIKTIVNO ODRŽAVANJE PO STANJU

Prediktivno održavanje je uvjetno vođen program preventivnog održavanja za kojega se još koristi izraz održavanje po stanju ili *CBM* (eng. *Condition-based maintenance* - *CBM*). Prediktivno održavanje sastoji se od redovitog praćenja stvarnog mehaničkog stanja, radne učinkovitosti i drugih pokazatelja radnog stanja raznih strojeva. Danas veliku ulogu u prediktivnom održavanju imaju senzori. Senzori prikupljaju podatke na temelju kojih se donose odluke u svrhu osiguranja maksimalnog intervala između popravaka te minimiziranje broja i troškova neplaniranih zastoja koji nastaju zbog kvarova [1]. Prediktivnim održavanjem moguće je poboljšati proizvodnost, kvalitetu proizvoda i cjelokupnu učinkovitost proizvodnje.

Prediktivne tehnologije moraju se koristiti kao alat za optimizaciju procesa. One služe kako bi otkrila, izolirala i pružila rješenja za sva odstupanja od prihvaćene izvedbe sustava koji može rezultirati izgubljenim kapaciteta, lošom kvalitetom, prevelikim troškovima ili opasnosti za sigurnost radnika.

Metode, postupci i sredstva koja se koriste kod prediktivnog održavanja, služe kako bi se kontinuiranim mjerenjem određenih parametara odredilo stanje opreme. Dobiveni rezultati se zatim uspoređuju s utvrđenim graničnim vrijednostima normalnog rada te se na temelju njih donose odluke o provođenju daljnjih akcija održavanja.



Slika 1. Opći slučaj određivanja i značenja granica mjernog parametra [2]

Slika 1 prikazuje način na koji se mogu prikazati izmjerene vrijednosti parametra, te na koji način je moguće prepoznati poremećaje u radu s obzirom na granice normalnog rada. Najčešće tehnologije održavanja po stanju su kontrola mjerenjem vibracija, kontrola temperature, vizualna kontrola, kontrola pukotina, kontrola mjerenjem buke, kontrola korozije i kontrola



podmazivanjem. Tvrtke same moraju izabrati metode održavanja, koje ovise o vrsti i tipu njihovog poslovanja, te kojim strojevima raspolažu.



**Slika 2. Postupak uvođenja održavanja po stanju kontrolom parametra [2]**

Nakon odabira metode mjerenja, mora se odabrati lokacija mjerenja, parametri te dopuštene vrijednosti prije samog početka mjerenja. Ako je dijagnoza pogrešna, potrebno je ponovno krenuti od početka i izabrati novu metodu mjerenja, lokaciju, parametre i dopuštene vrijednosti. Ako je dijagnoza dobra, potrebno je donijeti prikladan plan održavanja kojim je moguće očuvati sustav, smanjiti kvarove i samim time smanjiti troškove održavanja (Slika 2).

### 3. RAČUNALNI SUSTAV SA SPOSOBNOSTĆU PREDVIĐANJA STANJA OPREME

Kako bi sustavi postigli visoku pouzdanost, potrebno je provoditi djelotvorne aktivnosti održavanja kako bi se problemi mogli otkriti i ispraviti prije nego što oni postanu ozbiljni i uzrokuju zastoje u sustavima. Učinkovito provodeći strategiju održavanja po stanju, moguće je povećati pouzdanost te smanjiti troškove koji nastaju kao posljedica zastoja. Tijekom desetljeća razvijene su i implementirane mnoge strategije održavanja koje se sad većinom temelje na dijagnostičkim i prognostičkim sposobnostima. To je omogućeno razvojem senzora i tehnologijama praćenja sustava.

Kako je već spomenuto, takve strategije dio su prediktivnog održavanja odnosno CBM-a. Ono zamjenjuje unaprijed zadane zadatke održavanja samo s potrebnima i to na temelju trenutnog stanja opreme. Takva strategija mora ispuniti nove zahtjeve koji se pojavljuju u održavanju na način da će moći predviđati stanje sustava u budućnosti i donositi odluke u skladu s time [3].

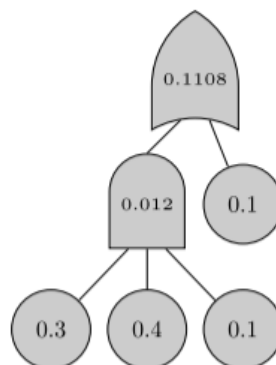
Istraživačka i inženjerska težnja jest razvoj i primjena sustava sa sposobnošću predviđanja stanja opreme. Računalni sustav sa sposobnostima procjene stanja opreme (eng. *Prognostics and health management - PHM*) se sastoji od naprednih dijagnostičkih, prognostičkih i istraživačkih aktivnosti koje omogućuju i podržavaju CBM. PHM privlači značajne istraživačke interese zbog sve veće potrebe za modelima predviđanja i odlučivanja, koje su sastavni dio za učinkovito obavljanje CBM strategije [5]. Razvojem Interneta stvari (eng. *Internet of things – IoT*) i analize velikog broja podataka uz pomoć naprednih tehnologija obrade podataka, sve je veća mogućnost primjene prediktivnih modela, ne samo u održavanju već u svim granama industrije. Kako se radi o fleksibilnom sustavu, koji doživljava veliki razvoj, PHM je postao potreba za trenutno aktualnu četvrtu industrijsku revoluciju koja se povezuje s IoT-om [6].

Provođenje PHM-a za kompleksne sustave može biti vrlo zahtjevno. Kako bi se to pojednostavilo, svaka komponenta se gleda zasebno i na svaku se posebno primjenjuje PHM. Takav pristup sastoji se od dva dijela. Prvi dio sastoji se od [5]:

1. Identifikacije kritičnih komponenti:

Jedan od mogućih načina otkrivanja kritičnih komponenti je korištenje analize opasnosti. Ona procjenjuje kolika je mogućnost da će se neki događaj ili stanje dogoditi. Za izračunavanje potrebno je veliko znanje tehničkih sustava koji se promatraju. Postoji nekoliko tehnika analize opasnosti koje se mogu podijeliti u dvije skupine: kvantitativna

i kvalitativna. Najpoznatija kvalitativna metoda je Analiza uzroka i posljedica kvarova (eng. *Failure mode and effects analysis – FMEA*). To je metoda koja analizira potencijalne kvarove opreme i njihove posljedice. Izračunava se vrijednost prioriteta rizika, koji se gleda kao umnožak ozbiljnosti, pojavljivosti i primjetljivosti rizika (kvara) [2]. Najpoznatija kvantitativna metoda, koja se koristi u kontekstu PHM-a je analiza stabla kvarova (eng. *Fault tree analysis – FTA*). FTA predstavlja faktore i događaje koristeći standardne logičke simbole.



**Slika 3. Primjer analize stabla kvarova [7]**

Slika 3 prikazuje primjer analize stabla kvarova. Brojevi prikazuju vjerojatnosti da će se neki događaj dogoditi. Gornji simbol prikazuje događaj koji se promatra te će se on dogoditi ukoliko se dogode bilo koji od sljedeća dva događaja (vjerojatnost 0,012 ili 0,1). Događaj s vjerojatnošću 0,012 dogodit će se koliko se dogode sva tri događaja koja predstavljaju ulaz (vjerojatnosti 0,3, 0,4 i 0,1).

## 2. Odabir fizičkih parametra za praćenje

Nakon pronalaženja kritičnih komponenata, potrebno je odrediti odgovarajuće fizičke parametre koji će se promatrati. Parametri se određuju na temelju prijašnjih iskustava. Za mehaničke sustave, pratit će se kroz duži vremenski period: pozicija, brzina, akceleracija, moment, vibracija, temperatura i naprezanje. Promjena bilo koje navedene stavke, može označavati promjenu u stanju komponente.

## 3. Odabir senzora za praćenje

Za odabir senzora, koji će služiti za promatranje sustava, potrebno je uzeti u obzir sljedeće stavke: parametri koji se prate, pouzdanost, točnost, mjerno područje, rezolucija, karakteristike i cijena. Kada su senzori postavljeni, stručnjaci počinju s prikupljanjem podataka.

## 4. Prikupljanje podataka.

Svi prikupljeni podaci kritičnih komponenti spremaju se na osobno računalo te se mogu podijeliti u dvije skupine. Prva skupina su podaci o događaju koji uključuju kvalitativne informacije o promatranoj komponenti poput opisa o instalaciji, kvarovi, popravci, uzroci i tako dalje. Također se sastoje od opisa popravka kvara i ozbiljnosti popravka. Drugi podaci, koji se prikupljaju, su podaci o praćenju stanja koji su povezani uz stanje komponente. To mogu biti podaci o vibracijama, o ulju, temperaturi, tlaku, vlazi i tako dalje.

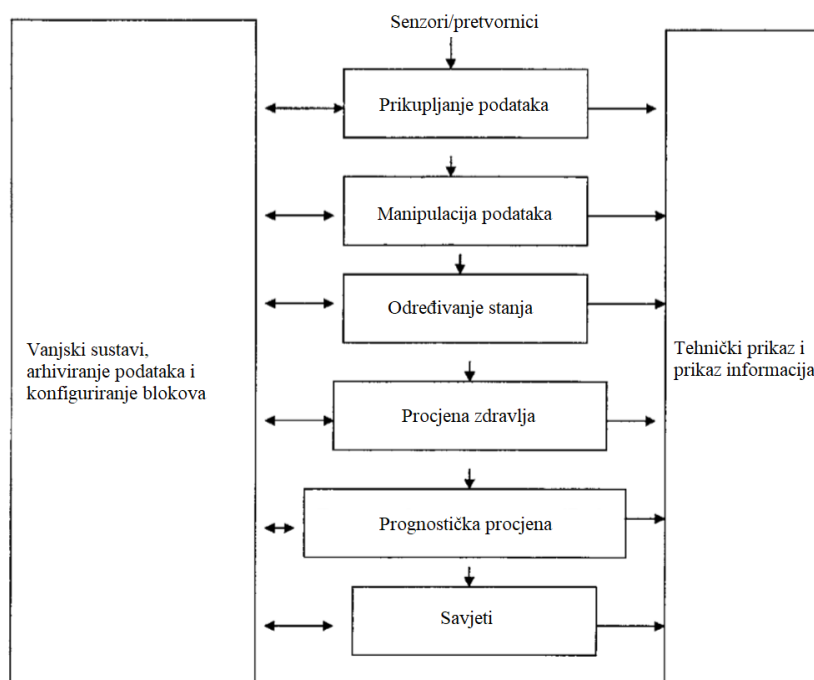
5. Predobrada signala sa senzora.

Zbog mogućih pogrešaka u signalima, koji nastaju zbog različitih vrsta buke, potrebno je obaviti predobradu podataka. Na taj način može se poboljšati kvaliteta ulaznih parametra i pripremiti signal za bolju analizu u kasnijim koracima.

U drugom dijelu, stručnjaci PHM-a i analitičari koriste dobivene podatke sa senzora kako bi provodili neke od sljedećih zadataka PHM-a [5]:

1. Analiza podataka,
2. otkrivanje grešaka,
3. dijagnostika,
4. prognoziranje,
5. donošenje odluka,
6. te izrada korisničkog sučelja.

Osim široke primjene PHM-a u industriji koja može značiti velike financijske uštede, ona se koristi i u područjima poput obrade signala, strojno učenje i rudarenje podataka [8]. Glavni pristup prognoziranja je na temelju podataka upravo zbog svoje intuitivnosti koje je moguće koristiti zbog senzora, spremanja i analiziranja podataka [5].



**Slika 4. Model obrade podataka prema ISO 13374-1 [3]**

Slika 3. prikazuje model obrade podataka koji je načinila Međunarodna organizacija za standardizaciju (eng. *International Organization for Standardization – ISO*), koji čini konceptualni okvir za obradu podataka s ciljem lakšeg razumijevanja i primjene CBM/PHM metoda od prikupljanja podataka i analize do procjene zdravlja, prognostičke procjene i savjeta [3].

### 3.1. Senzori

Senzori predstavljaju osnovu za dijagnostiku i prognoziranje kvarova. Pomoću njih se prikupljaju podaci koji će se zatim obraditi pomoću određenih algoritama. Uspješnost algoritama ovisit će o vrsti, broju i lokaciji senzora. Za svaki sustav postoje različiti senzori, te oni prvenstveno ovise o parametru koji se želi izmjeriti. Pri odabiru senzora, također je potrebno, uzeti u obzir njihovu veličinu, težinu, cijenu i druga svojstva.

Podaci prikupljeni sensorima, rijetko se mogu koristiti u izvornom obliku. Potrebno ih je obraditi na odgovarajući način kako bi se sačuvale karakteristične značajke koje ukazuju na kvarove koji se žele otkriti. Takvi podaci će se filtrirati, umanjiti i ispraviti kako bi se smanjila razina buke i volumen podataka koji se onda naknadno obrađuje.

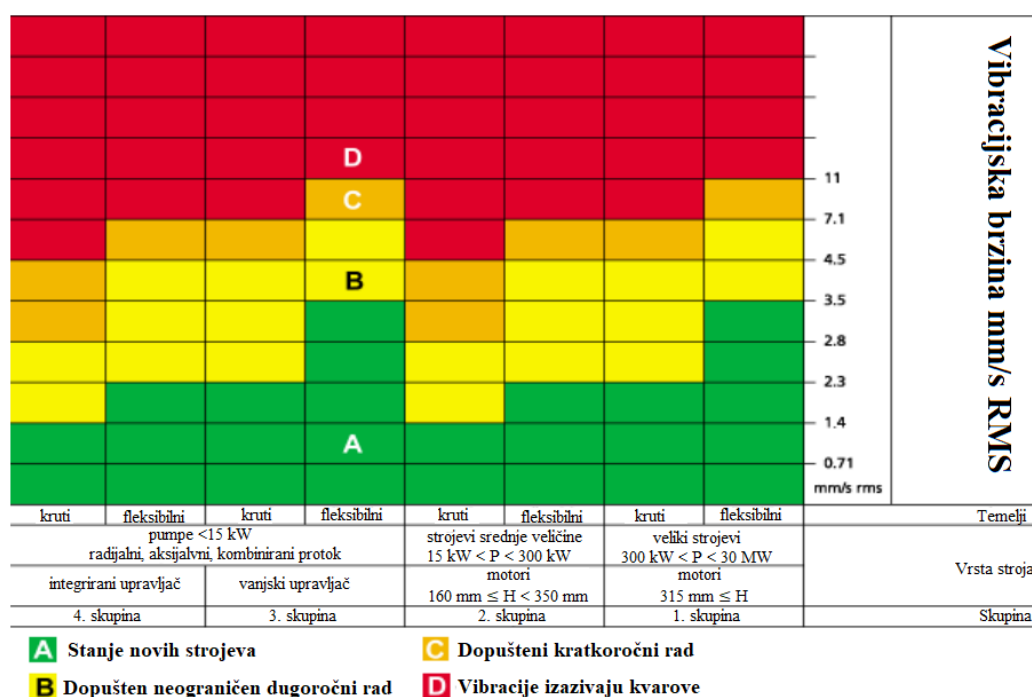
Senzori imaju specifično područje primjene, a namijenjeni su za praćenje parametra o stanju kao što su temperatura, tlak, brzina, vibracije i tako dalje. Neki senzori su posebno umetnuti za mjerenje količina koje su izravno povezane s identificiranjem načina kvarova. Među njima su

instrumenti za mjerenje naprezanja, ultrazvučni senzori, uređaji za blizinu, senzori za akustičnu emisiju, senzori za elektrokemijski zamor, interferometri, i tako dalje [3].

### 3.1.1. Mjerenje vibracija

Sva mehanička oprema u gibanju vibrira određenom frekvencijom te je tu frekvenciju moguće izmjeriti u svakom trenutku rada opreme. S obzirom da je poznata frekvencija normalnog rada opreme, moguće je zabilježiti njezine promjene. Uzrok promjena mehaničkog stanja opreme mogu biti: trošenje, promjene u okolini sustava (npr. buka, curenje ulja), varijacije u opterećenjima. Kada se koriste ispravno, kontrola i analiza vibracija omogućuju održavanje optimalnih radnih uvjeta te učinkovitosti kritičnih dijelova sustava. Dobro poznavanje dinamike strojeva i kako sile stvaraju određene vibracijske frekvencije su ključni dio za razumijevanje uzroka vibracija [1].

Analiza vibracija je široko korištena tehnika održavanja po stanju, koja omogućuje otkrivanje promjena stanja opreme u ranoj fazi.



Slika 5. Procjena vibracijskog stanja sustava prema ISO 10816 [4]

Kako bi se olakšala procjena vibracijskog stanja, postoji norma koja utvrđuje opće uvjete i postupke mjerenja i vrednovanja vibracija pomoću mjerenja izvedenih na nerotirajućim dijelovima strojeva [3]. Slika 3 prikazuje ISO 10816 standard za vibracijske brzine različitih vrsta i veličina strojeva.

Senzori za vibracije prate vibracijsku karakteristiku prisutnu tijekom rada stroja te se mogu smatrati najboljom početnom točkom pri razvoju prediktivne strategije. Senzore je potrebno montirati na kruti dio stroja i što bliže predmetu koji se mjeri (ležaj, osovina, i tako dalje) kako bi se osigurao pravilan prijenos signala vibracija. Senzori za vibracije otkrivaju ukupno povećanje vibracija koje može ukazivati na kvar pojedinog dijela stroja koji se promatra [2].



**Slika 6. Primjer sustava za mjerenje vibracija ležajeva SPM – Leonova i Brüel & Kjær [2]**

Na slici 6, prikazana su dva primjera uređaja za mjerenje vibracije opreme strojeva od tvrtki Leonova i Brüel & Kjær.

### **3.1.2. Kontrola temperature**

Temperatura se može mjeriti izravno ili uz pomoć odzračene energije u infracrvenom spektru. Za primjenu izravnog mjerenja koriste se detektori otpora temperature (eng. *Resistive Temperature Detectors – RTD*) i termoparovi. Za indirektna mjerenja koriste se beskontaktni infracrveni senzori. Primjenjuju se za temperature ležajeva, podmazivanja i temperaturu rashladne vode [3].

Termografija je tehnika prediktivnog održavanja koja se koristi za praćenje temperature strojeva, konstrukcija i sustava postrojenja. U tehnici se koriste uređaji koji prate emisiju infracrvene energije za određivanje radnog stanja. Otkrivanjem toplinskih promjena (područja koja su toplija ili hladnija nego što bi trebala biti) moguće je locirati i odrediti probleme unutar postrojenja. Uređaji koji se koriste za kontrolu temperature su infracrveni termometri, linearni skeneri i sustavi za prikaz slika [1].

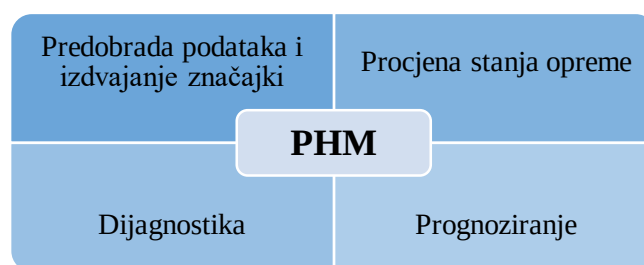
### **3.1.3. Kontrola podmazivanjem**

Postoji nekoliko metoda na kontrolu podmazivanja koje se koriste kod prediktivnog održavanja, a to su: analiza ulja za podmazivanje, spektrografska analiza, ferografija i analiza čestica trošenja.

Analiza ulja za podmazivanje je tehnika koja određuje stanje ulja za podmazivanje u mehaničkoj i električnoj opremi. Ne služi za određivanje radnog stanja strojeva. Tom analizom moguće je odrediti da li ulje zadovoljava zahtjeve za podmazivanje stroja. Primarne primjene prethodnih metoda su kontrola kvalitete, smanjenje zaliha ulja za podmazivanje i određivanje najisplativijeg intervala za zamjenu ulja [4].

### 3.2. Analiza računalnog sustava sa sposobnošću predviđanja stanja opreme

Za primjenu PHM-a potrebna su znanja iz raznih područja poput računalstva, obrađivanja signala, statistike, kao i ekspertnih znanja za određenu opremu. Postoje četiri glavna zadatka za analiziranje podataka i modeliranje PHM-a: predobrada i izdvajanje značajki, procjena stanja opreme, dijagnostika i prognoziranje (slika 7).



Slika 7. Zadaci u analizi PHM sustava [8]

Prije obavljanja tih zadataka potrebno je izvesti radnje koje su objašnjene na samom početku ovoga poglavlja, a sastoje se od identifikacije kritičnih komponenti na koje će se prioritzirati primjena PHM-a, odabira parametra te prikupljanje podataka [8].

#### 3.2.1. Predobrada podataka i izdvajanje značajki

Zadatak predobrade i izdvajanja značajki uključuje procjenu kvalitete podataka, čišćenje podataka, identifikaciju režima i segmentaciju. Iako ne nudi izravne informacije koje se mogu odmah primijeniti, to je ključan korak i zahtijeva poznavanje područja i vještine obrade podataka za održavanje vrijednih dijelova podataka prilikom uklanjanja neželjenih komponenti. Provjera kvalitete podataka mora osigurati da ne postoje pogreške u senzorima niti prikupljenim podacima.

Tablica 1. Metode za predobradu podataka [8]

| Metoda                        | Prednosti                                                   | Nedostaci                                                                            |
|-------------------------------|-------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1 Provjera kvalitete podataka | Potrebno je poznavati tipove signala, posebice za vibraciju | Potrebni su pragovi kako bi se moglo odlučiti da li će se signal uključiti u analizu |



|          |                       |                                                                               |                                                                              |
|----------|-----------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| <b>2</b> | Identifikacija režima | Važna je za razvoj osnovnih skupova podataka za svako operativno stanje       | Potrebne su naprednije metode jer se često mijenja operativno stanje sustava |
| <b>3</b> | Uklanjanje anomalija  | Odstupajuće, konstantne i nedostajuće vrijednosti, mogu izazvati lažne alarme | Potrebna su posebna znanja za otkrivanje odstupajućih vrijednosti            |

U tablici 1 prikazane su neke od metoda koje se koriste u predobradi podataka.

Također postoje brojne metode za izdvajanje značajki ili karakteristika izmjerenih signala, a neke od njih su nabrojane u tablici 2.

**Tablica 2. Metode za izdvajanje značajki [8]**

|          | Metoda                                                 | Prednosti                                                                                               | Nedostaci                                                                                              |
|----------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>1</b> | Metode izdvajanja značajki na temelju frekvencija      | Obrada domena frekvencije omogućava izdvajanje specifičnih kvarova komponenti                           | Zahtjeva veću stopu uzrokovanja i veće troškove prikupljanja podataka                                  |
| <b>2</b> | Metode izdvajanja značajki na temelju ostatka          | Prikladnije za niske frekvencijske signale i signale s potencijalnom korelacijom                        | Algoritmi ponekad zahtijevaju treniranje neuronske mreže što zahtjeva više programiranja               |
| <b>3</b> | Statistika za svaki dio procesa ili vremenski odsječak | Idealna za procesne signale i omogućuje jednostavan način hvatanja ključnih dijelova izmjerenog signala | Potrebne su informacije o kontekstu za prepoznavanje različitih vremenskih odsječaka procesnog signala |
| <b>4</b> | Statistika vremenskih serija                           | Zahtjeva najmanje znanje o domeni te se najlakše primjenjuje                                            | Pruž a manje informacija od ostalih metoda                                                             |

Metoda izdvajanja značajki na temelju frekvencija koristi se za visokofrekventne signale poput vibracija ili struje. Za primjenu u kojemu se promatrani signal sastoji od tragova signala poput temperature, tlaka ili slično, preporučuje se drugačiji način izdvajanja značajki, a ono se bazira na temelju ostatka. Za takav tip algoritma, osnovu čine normalno operativno stanje stroja. Ono se zatim koristi za uspoređivanje predviđenih vrijednosti senzora.

Izdvajanje statističkih parametara jednostavan je i učinkovit način za karakterizaciju dostupnih signala regulatora. Određivanje statistike tijekom različitih vremenskih odsječaka može predstavljati drugačije kretanje ili radnju koje provodi nadzorni sustav [8].

### 3.2.2. Procjena stanja opreme

Zadatak procjene stanja opreme sastoji se od procjene i kvantificiranja stanja opreme analizom prikupljenih podataka. Ako postoje podaci o neispravnom stanju, tada se može izračunati vjerojatnost kvara komponente. Međutim, ako podaci o kvarovima komponenti nisu dostupni, procjena stanja opreme može se pretvoriti ili u problem praćenja degradacije za postupne kvarove ili u problem otkrivanja kvarova za iznenadne kvarove.

Tablica 3 prikazuje najčešće korištene algoritme kod procjene stanja opreme.

**Tablica 3. Algoritmi za procjenu stanja opreme [8]**

| Metoda                                    | Prednosti                                                                                               | Nedostaci                                                                                          |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>1</b> Mjerenje kombinacije značajki    | Jednostavna primjena, lakše postavljanje pragova na temelju raspodjele stanja opreme                    | Ne postoji korelacijski odnos između značajki                                                      |
| <b>2</b> Udaljenost od normale            | Potreban je osnovni skup podataka za treniranje algoritma, mogući je korelacijski odnos između značajki | Ne uzima u obzir da li su značajke niže ili veće od očekivanih                                     |
| <b>3</b> Testiranje statističkih hipoteza | Jednostavna primjena, moguće je odrediti da li sustav normalno radi                                     | Podaci možda ne odgovaraju pretpostavljenoj distribuciji za testiranje hipoteza                    |
| <b>4</b> Regresijska metoda               | Pruža mapiranje značajki oštećenja na izlazu ili vrijednosti stanja                                     | Zahtjeva izlaznu vrijednost koja je povezana sa stanjem sustava i više skupova podataka za trening |
| <b>5</b> Jednorazredni klasifikator       | Algoritmi za opis vektorskih podataka mogu pružiti granice za određivanje anomalija                     | Potrebno je iskustvo pri odabiru parametara i funkcije kernela                                     |

Najjednostavniji pristup za procjenu stanja opreme je mjerenje kombinacije značajki gdje se izračunava statistički prag za otkrivanje degradacije koje je moguće dobiti pomoću distribucije

vrijednosti stanja. Udaljenost od metrike za normalno stanje može se koristiti za procjenu stanja sustava ili opreme. Za jednostavno otkrivanje anomalija koristi se testiranje statističkih hipoteza poput testa omjera vjerojatnosti ili *T-test*. Još jedna metoda za otkrivanje anomalija je jednorazredni klasifikator poput algoritma podataka potpornih vektora (eng. *Support vector data – SVD*) koji je detaljno objašnjen u poglavlju 4.2.1. Regresija ili neuronske mreže koriste se kada postoji dovoljno podataka za razvoj regresijskog modela. Takav model može se koristiti kod mapiranja između vrijednosti značajki i vrijednosti stanja ili veličine kvara [8].

### **3.2.3. Dijagnostika**

Dijagnostika se odnosi na klasifikaciju različitih načina kvarova izdvajanjem grešaka iz podataka. Moguće je koristiti različite značajke zajedno s klaster algoritmom ili algoritmom za klasifikaciju za razvoj modela temeljenog na podacima za dijagnostiku kvarova opreme[8].

Mogućnost dijagnostike kvarova vrlo je složeno područje PHM-a i nalazi na mnoge izazove. Samo otkrivanje kvarova već kod njihova stvaranja još uvijek je limitirano. Razlozi tomu su velika osjetljivost na buku u signalu, ovisnost o okolini i radnim uvjetima, manjak otkrivanja kvarova i nesigurnost u rasporedu održavanja. Dijagnostika održavanja uglavnom je specijalizirani proces, tako da je u svakoj industriji taj proces drugačiji. Dijagnostika ima veliku važnost za uspješno prognoziranje jer prihvatljiva prognostička metoda započinje robusnom dijagnostikom, budući da neizvjesnosti procijenjenog sustava mogu utjecati na buduća predviđanja. Izazovi za dijagnostiku postoje zbog problema provjere i valjanosti te mogu postojati velike netočnosti zbog kompleksnosti sustava. Dijagnostičke i prognostičke metode moraju biti sposobne nositi se s neizvjesnostima. Ako im se ne posveti, te neizvjesnosti mogu uzrokovati lažne alarme, netočna predviđanja i samim time krive odluke [9]. S obzirom na to da će u okviru ovog rada biti korišteni algoritmi metode za dijagnostiku sustava odnosno otkrivanje kvarova, oni će biti detaljno objašnjeni u narednom poglavlju (4.).

### **3.2.4. Prognoziranje**

Zadatak prognoziranja je određivanje zdravstvenog stanja imovine. Ako je poželjno kratkoročno predviđanje, često se koristi modeliranje vremenskih serija kako bi se previdjelo kada će se stroj pokvariti. Ako je poželjno dugoročno predviđanje, tada se predviđa preostali korisni vijek trajanja s mnogim postojećim alatima za strojno učenje i statistiku [8].

Prognoziranje se susreće s još više izazova od dijagnostike i to je jedan od glavnih razloga zašto je prognoziranje još uvijek nerazvijeni dio PHM sustava. Jedan od glavnih problema je što su neki kvarovi povremeni i stoga ih je teško predvidjeti. Područje prognoziranja još uvijek je u procesu razvoja te se trenutne tehnologije ne smatraju dovoljno dobre jer ne računaju

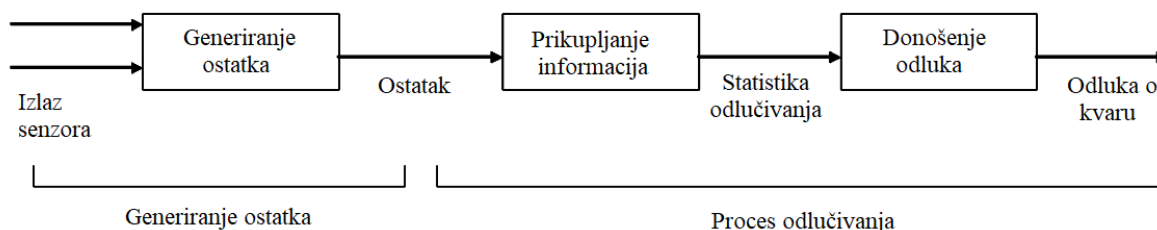
nesigurnosti, slabe su im provjere i valjanosti sustava te ne postoji procjena rizika za razvoj PHM sustava. Ne postoji standardi rok procjene prognostičkih algoritama. Potrebne su metode koje će kvantificirati točnost tehnologija za prognostičke procjene.

Prognoziranje je zapravo procjena preostalog korisnog života (eng. *Remaining useful life* – *RUL*), na temelju uvjeta, kako bi se donijele bolje odluke u održavanju. RUL je predviđanje vremena ili ciklusa prije nego funkcioniranje proizvoda ili procesa dosegne neprihvatljiv prag. Bez odgovarajuće mjerne nesigurnosti, RUL ima malu vrijednost. Možda je glavni izazov za prognoziranje to što će uvijek postojati neki limit u točnosti i preciznosti kod određivanja RUL-a zbog same prirode koja je predviđanje sustava [9].

## 4. PRAĆENJE I DIJAGNOSTIKA KVAROVA

Kako je navedeno u 3. poglavlju, dijagnostika je samo jedan od zadataka koji se primjenjuje u sklopu PHM-a. S obzirom na sve veću kompleksnost industrijske opreme i sustava, sve manje prostora ostaje za smanjenje proizvodnosti, učinkovitosti te sigurnosti. To predstavlja povećane zahtjeve za praćenje i dijagnostiku kvarova u realnom vremenu.

Kvar se definira kao najmanje nedopušteno odstupanje barem jednog karakterističnog svojstva ili parametra sustava od prihvatljivog, uobičajenog ili standardnog stanja [10]. Dijagnostika kvarova bavi se otkrivanjem, izoliranjem i identificiranjem predstojećih ili početnih stanja kvarova koji zahvaćaju komponente koje su još uvijek u pogonu i rade, ali s manjom učinkovitosti. Cilj je postići najbolju moguću izvedbu zadane dijagnostike, tako da se minimiziraju lažni pozitivni i negativni rezultati uz istovremeno smanjenje vremenskog kašnjenja pokretanja i otkrivanja događaja kvara [3]. Slika 6 prikazuje opću strukturu i zadatke koji se provode kako bi se mogli otkriti i identificirati kvarovi.



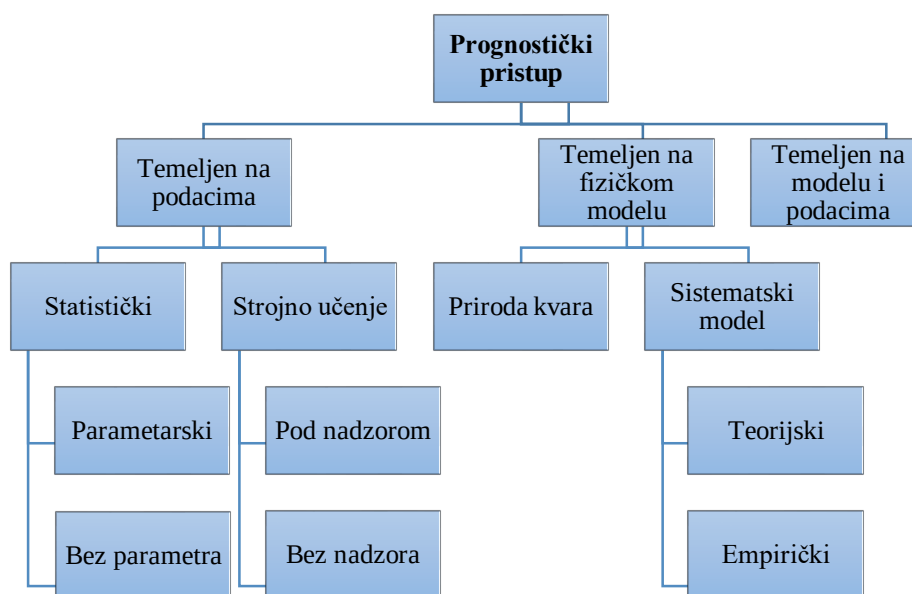
**Slika 8. Opća struktura otkrivanja i identifikacije kvarova [3]**

Na ovoj slici, preostali signal koji predstavlja odstupanje od standardnih uvjeta, generira se, na primjer, usporedbom izlaznog modela sa stvarnim izlazom sustava. Na temelju tog signala, donosi se odluka o radnom stanju stroja. Metode za generiranje ostatka mogu se temeljiti na statistici vremenskog signala, identifikaciji sustava, tehnikama frekvencijske domene, i tako dalje [3].

Dijagnostika kvarova sastoji se od tri zadatka [10]:

1. Otkrivanje kvarova se koristi za utvrđivanje da li postoje neispravnosti ili kvarovi u sustavu te za određivanje vremena događanja kvara,
2. izoliranje kvarova se koristi za određivanje lokacije neispravnih komponenti.
3. identificiranje kvarova se koristi za određivanje vrste, oblika i veličine kvara.

Tehnologije, koje omogućuju ovakav način dobivanja podataka i donošenje odluka dijele se u dvije kategorije: temeljene na fizičkom modelu i temeljene na podacima. Takve tehnologije odnosno pristupi mogu se podijeliti u još nekoliko razreda (Slika 9.).



Slika 9. Klasifikacija sustava za predviđanje stanja opreme [11]

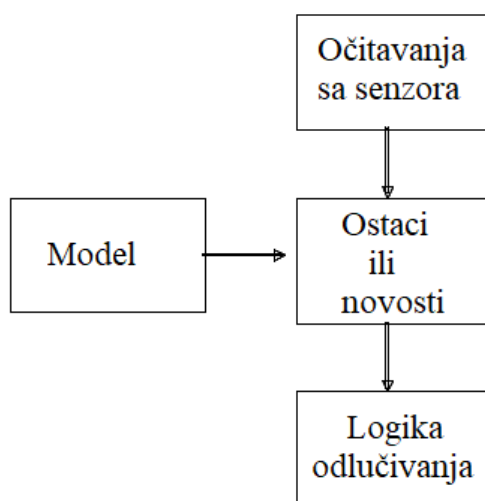
#### 4.1. Dijagnostika kvarova temeljena na fizičkim modelima

Za tehnologije temeljene na fizičkom modelu potrebno je imati modele industrijskih procesa ili praktičnih sustava koji se mogu dobiti bilo fizičkim principima ili tehnikama identifikacije sustava. Na temelju fizičkih modela, razvijaju se algoritmi za dijagnostiku kvarova za nadgledavanje usklađenosti između izmjerenih izlaza promatranih sustava i modela koji predstavljaju predviđeni izlaz [10]. Razlike između stvarnih i predviđenih ponašanja smatraju se indikatorima kvarova. Oni se još nazivaju ostatci te ih je moguće dobiti na nekoliko sljedećih načina [15]:

- S obzirom da je sustav moguće promatrati (na primjer, moguće je odrediti ponašanje cijelog sustava na temelju izlaza sustava) te su poznati parametri, promatrač može procijeniti proces, a ostaci se mogu generirati rekonstrukcijom nepoznatog unutarnjeg stanja sustava. Ne postoji točan i potpuni fizički model procesa u praksi, te je potrebno uzeti u obzir nepouzdanost modela koja utječu na ostatke koji se mogu interpretirati kao neispravne situacije.
- Paritetni prostor zasnovan je na transformaciji stanja prostornog modela postrojenja kako bi se dobili paritetni odnosi promatranjem sustava na konačnom horizontu. Svrha stvaranja paritetnih odnosa je osiguranje jednadžbi koje ovise samo o poznatim ili izmjerenim varijablama ulaza i izlaza. Za dobivanje odnosa pariteta koriste se redundantni između različitih varijabli.

- Procjena parametra za otkrivanje i dijagnostiku kvarova temelji se na usporedbi nominalnih parametara izračunatih u radu bez kvarova i parametra procijenjenih tijekom određenog vremenskog razdoblja. Kvar se otkriva kada procijenjene vrijednosti sustava nisu u skladu s nominalnim.
- Bond-ov graf obuhvaća različite vrste informacija zbog svojih uzročno strukturnih značajki koje omogućuju izravno stvaranje niza indikatora kvarova. Graf je proširen uključivanjem parametra nesigurnosti kako bi se stvorio čvrst i prilagodljiv prag za fazu ocjene ostataka. Postupak dobivanja ostataka temelji se na pokrivanju uzročnih staza i provodi se u softverima.

Slika 10 prikazuje način rada na kojemu se provodi dijagnostika temeljena na fizičkom modelu.



**Slika 10. Tehnologije dijagnostike kvarova temeljene na fizičkom modelu [3]**

#### 4.2. Dijagnostika kvarova temeljena na podacima

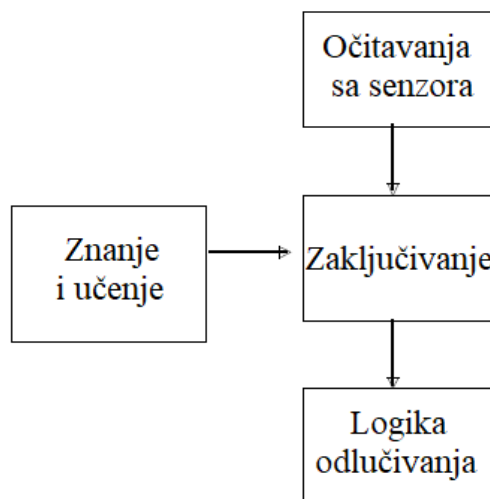
Pristupi temeljeni na podacima ne zahtijevaju modele niti posebna znanja sustava kako bi započeli prognoziranje. Snimani i povijesni podaci koriste se za učenje o ponašanju sustava te kako bi se izvodile dijagnostika i prognoziranje. Stoga je ovakav pristup prigodan i za složene sustave s ponašanjima koji se ne mogu procijeniti, a mogu promijeniti. Primjena tehnika temeljenih na podacima u svrhu praćenja i prognoziranja stanja sustava temelji se na pretpostavci da se statističke karakteristike izvedbi sustava neće promijeniti dok se ne dogodi kvar. Stoga je glavna prednost ovakvog pristupa što se algoritmi mogu brže postaviti i učinkovitiji su kod korištenja, za razliku od drugih tehnologija. Međutim potrebno je imati povijesne podatke i znanje o podacima o operativnoj učinkovitosti. Tehnologije temeljene na

podacima u potpunosti se oslanjaju na analizu podataka dobivenih iz senzora i iskorištavanja operativnih signala ili signala povezanih s izvedbama koje mogu ukazivati na stanje sustava koji se nadgledava [11].

Glavni nedostatak ovog pristupa je što ovisi o dostupnosti povijesnih i empiričkih podataka. Ponekad je teško skupiti podatke za tek nove sustave ili komponente, te je za to potrebno vrijeme, a i može biti skupo. Postoje tehnologije i procedure koje ipak to mogu omogućiti [11]:

1. Hardver u petlji simulacije (*eng. Hardware-in-the-loop – HiL*) je računalna simulacija koja se koristi za testiranje hardvera pod simuliranim opterećenjima kao u stvarnoj primjeni. Nekoliko parametra kvara može se kontrolirati neovisno. HiL se također koristi i za provjeru algoritma, benchmarking i razvoj metrike za prognoziranje.
2. Ispitivanje ubrzanog života (*eng. Accelerated life test – ALT*) konstruirano je tako da uzrokuje kvar na proizvodu mnogo brže nego u normalnim uvjetima primjenom povećanih naprezanja koja uzrokuju otkazivanje mehanizma. Moguće je primijeniti nekoliko okolišnih stanja i opterećenja zasebno u svrhu bržeg kvarenja sustava.
3. Online učenje se temelji na pretpostavci da se novi sustavi ne kvare određeni period. Taj period je najčešće na početku rada, te se oni uzimaju kao normalno stanje sustava.

Slika 11 prikazuje način rada tehnologija koje se temelje na podacima.



**Slika 11. Tehnologije dijagnostike kvarova temeljene na podacima [3]**

#### 4.2.1. Statistički pristup

Statistički pristup koristi statističke parametre, poput srednje vrijednosti, varijance, medijana i tako dalje, da bi predvidjela vjerojatnost na temelju poznate ili nepoznate temeljne vjerojatnosne distribucije.



Statistički pristupi općenito se smatraju jednostavnim ako je poznato statističko svojstvo odnosno distribucija vjerojatnosti. Ovakav tip pristupa naziva se parametarski pristup. Tehnike procjene statističkih parametara i testiranje hipoteze mogu se primijeniti u otkrivanju anomalija u podacima. Nakon što je definirano normalno stanje sustava odnosno normalni operativni podaci putem distribucije vjerojatnosti, moguće je nove podatke klasificirati korištenjem različitih metoda poput [11]:

- Testiranje hipoteza jedan je od najjednostavnijih statističkih postupaka u kojemu je moguće provjeriti dolaze li podaci iz iste populacije kao i podaci korišteni prvi treniranju odnosno učenju. Također je moguće provjeriti da li je srednja vrijednost uzoraka ista kao i srednja vrijednost populacije ukoliko je poznata standardna devijacija.
- Analiza varijanci je metoda u kojemu se analizira srednja vrijednost nekoliko grupa uzoraka na koje mogu utjecati nekoliko različitih faktora.
- Procjena najveće vjerojatnosti je metoda za procjenu najvjerojatnije vrijednosti povezane s drugim vrijednostima u podacima o populaciji.
- Pomoću metode procjene *a posteriori* mogu se procijeniti parametri procesa ili sustava na temelju prijašnjih znanja o sustavu.

Bezparametarski pristup se ne temelji na pretpostavkama za statističkim parametrima određene populacije. Ovakav pristup je prikladan kada osnovna distribucija vjerojatnosti nije poznata i kada se podaci ne mogu modelirati standardnom raspodjelom. Neke od metoda koje ne koriste parametre su [11]:

- *Parzen-Window* procjena gustoće je metoda pomoću koje se gustoća procjenjuje kao da svi podaci za promatranje pripadaju funkciji koja može pridonijeti procjeni gustoće na temelju odabrane kernel funkcije.
- Histogram je najjednostavnija metoda koja uključuje dva koraka. Prvi korak je izrađivanje histograma na temelju dostupnih podataka koji najčešće predstavljaju normalno operativno stanje, a drugi korak je uspoređivanje novih podataka s izrađenim histogramom.
- Metoda najbližeg susjeda zahtjeva širinu koja opisuje poziciju jednog podatka u odnosu na drugi podatak. Takva metoda pretpostavlja normalnu operativnu radnju u gustim naseljima, dok se anomalije javljaju daleko od svojih najbližih naselja.

#### 4.2.2. Strojno učenje bez nadzora

Strojno učenje dijagnosticira kvarove na temelju prikupljenih podataka tako što pretvara prikupljene podatke u korisne informacije, koje se koriste zajedno s podacima prikupljenih sa senzora, kako bi omogućili buduća predviđanja. Strojno učenje se koristi u mnogim područjima poput prepoznavanje glasa, računalni vid, povrat informacija, robotika, medicinska dijagnostika, financijska predviđanja, praćenje cilja, biološka predviđanja, i tako dalje.

Ovakav pristup može se koristiti kao primjena u PHM-u tako što je moguće naučiti kako se sustavi ponašaju te raditi predviđanja na temelju generiranih predviđenih modela. Problemi PHM-a mogu se ili grupirati ili klasterirati pomoću strojnog učenja. Takvi podaci se zatim klasificiraju kao normalno stanje ili kao anomalije i mogu se poslije koristiti za izdvajanje anomalija i kvarova. Dalje se ove informacije mogu koristiti u programu predviđanja kako bi se predvidio preostali korisni život sustava.

Strojno učenje bez nadzora koristi se kada ne postoje dostupne označene varijable, odnosno podaci sa snimljenim i označenim tipovima kvara. Iako se može koristiti i u području otkrivanja anomalija, to jest kvarova, strojno učenje bez nadzora sve veću primjenu nalazi u području dijagnostike za automatizirano otkrivanje sličnih grupa unutar podataka na temelju tehnika klasteriranja ili procjene distribucije podataka unutar ulaznog prostora.

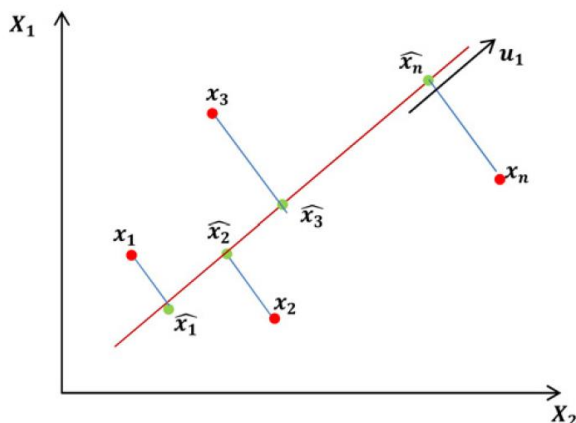
Može se također koristiti za mapiranje visokodimenzionalnog ulaznog prostora u niskodimenzionalni prostor u svrhu vizualizacije. Za većinu novih sustava, samo će normalni operativni podaci biti dostupni te će oni činiti jednu klasu, koju je moguće prepoznati tako da se naprave klasteri, to jest odredi se vektor koji će činiti tu klasu. Ti podaci se koriste kako bi se otkrilo stanje novih podataka na temelju razlike između trenutno naučenih klastera i klastera novih podataka. Drugim riječima, uz pomoć tehnika nenadziranog učenja moguće je označiti slične podatke i koristiti ih u danjoj dijagnostici.

Metode koje se koriste za strojno učenje bez nadzora su [11]:

1. Analiza glavnih komponenti

Analiza glavnih komponenti (eng. *Principal component analysis – PCA*) se koristi za smanjenje dimenzionalnosti, kompresiju podataka, izdvajanje značajki i vizualizaciju podataka mapiranjem podataka u niži dimenzionalni linearni prostor koji se naziva glavni potprostor. Cilj je smanjenje redundantnosti podataka, na način da se što manje informacija izgubi, kako bi se postiglo smanjenje dimenzionalnosti. Ovo je jednostavna metoda koja nema ograničenja parametra te se zbog toga koristi u raznim područjima.

Slika 12 prikazuje primjer PCA u kojemu se dvodimenzionalni podaci pretvaraju u jednodimenzionalne podatke minimiziranjem sume kvadrata projektiranih pogrešaka.



**Slika 12. Mapiranje dvodimenzionalnih podataka u jednodimenzionalne pomoću PCA [11]**

Tipični statistički podaci koji se koriste zajedno s PCA-om kod otkrivanja kvarova su [12]:

- $T^2$  predstavlja varijacije amplitudne projekcije opažanja u glavnom podprostoru. Relativno mala promjena, zbog pojave grešaka, bit će skrivena zbog velike količine varijabli koje su prirodno prisutne u glavnom prostoru. Tako će nastati brojni lažni alarmi. Utvrđeno je da  $T^2$  može otkriti samo kvarove koji utječu na nezavisne varijable. Takav rezultat podcjenjuje mogućnost otkrivanja kvarova glavnog podprostora, jer taj podprostor sadrži najrelevantnije podatke.
- Kvadratna pogreška procjene (eng. *Squared prediction error – SPE*) se bavi otkrivanjem kvarova na korelacijskim varijablama. Predstavlja trenutne razlike u zaostalom podprostoru. Tako pokazuje važne fluktuacije oko praga kad god se dogodi kvar. Ne odražava amplitudu kvara, jer je osjetljiva na razinu signala pri svakom promatranju. Procjena amplitude kvara na temelju načela rekonstrukcije nije točna, jer pretpostavlja da je SPE za bezobzirna opažanja nula. SPE, gotovo uvijek, ima vrijednost nula zbog buke prisutne u zaostalom prostoru. Kao posljedica, teorijska procjena je u velikoj mjeri pristrana.

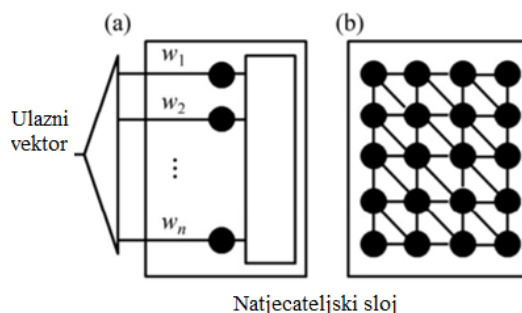
## 2. Klasteriranje

Klasteriranje je grupiranje podataka u  $K$  broj klastera tako da je udaljenost među točkama unutar klastera mala u usporedbi s udaljenosti točaka izvan klastera. Svaka sredina klastera dobiva oznaku  $\mu_k$  gdje je  $k=1,2,\dots,k$ . Točke podataka trebaju biti

dodijeljene tim klasterima tako da je zbroj kvadrata udaljenosti svake podatkovne točke od najbližeg centra najmanje jedan.

### 3. Neuronske mreže

Neuronske mreže također se mogu koristiti i u učenju bez nadzora gdje podaci s oznakom nisu dostupni. Samoorganizirajuće mape (eng. *Self-organizing maps – SOM*) vrsta su neuronskih mreža koje se koriste za učenje bez nadzora. Pomoću njih se višedimenzionalni prostor mapira u dvodimenzionalni prostor, na način da je očuvan prvobitni redoslijed. SOM se sastoji od dva sloja: ulazni sloj s ulaznim čvorovima i sloj s neuronskim jedinicama koje se nazivaju Kohonenove jedinice. Težinski vektor dodijeljen je svakoj vezi, od ulaznog sloja do Kohonenove jedinice. Kohonenova jedinica je u natjecateljskom sloju organizirana kao pravilna geometrijska struktura poput dvodimenzionalnog prostora.



**Slika 13. Primjer strukture SOM-a [13]**

Slika 13 (a) prikazuje opću strukturu SOM-a u kojemu je ulazni vektor povezan s Kohonenovim jedinicama preko težinskih vektora ( $w_1, w_2, \dots, w_n$ ). Na slici (b) je prikazana povezanost između Kohonenovih jedinica.

### 4. Kalmanovi filtri

Kalman je predložio tehniku za rješavanje problema kao što su: predviđanje slučajnog signala, odvajanje slučajnog signala od slučajne buke i otkrivanje signala poznatog oblika u prisutnosti slučajne buke. Kalmanovi filtri često se koriste kao optimizirana tehnika procjene stanja sustava. Rekurzivni je pristup za procjenu stanja sustava na temelju prethodnog poznavanja stanja sustava i izmjerenih informacija. Kalmanovi filtri također se koriste za spajanje mjerenja za istu varijablu iz različitih senzora.

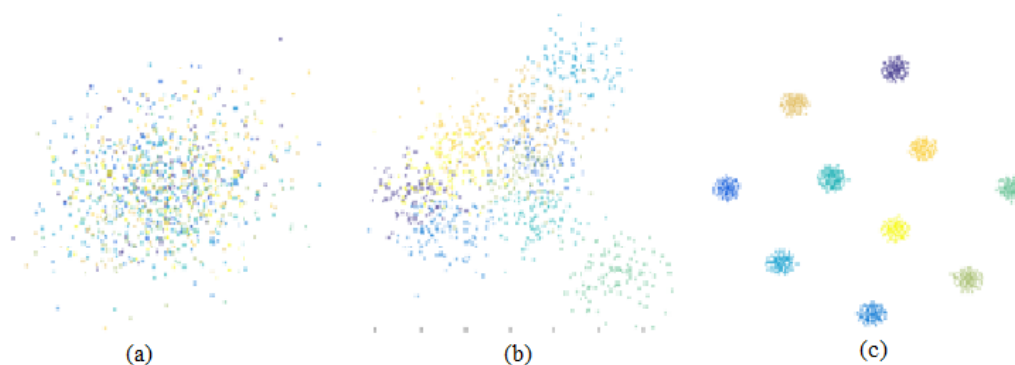
### 5. Filter čestica

Također se nazivaju i sekvencijalni Monte Carlo, a koriste se za obradu nelinearnosti modela ili promatračke buke. Razvijen je na temelju koncepta uzastopnog važnog

uzrokovanja i Bayesove teorije. Primjenjuje se u mnogim područjima poput ekonomije, biostatistike, praćenje ciljeva, analize vremenskih serija, obrade signala, i tako dalje.

#### 6. t-SNE algoritam

t-SNE (eng. *t-Distributed Stochastic Neighbor Embedding*) je nelinearni algoritam koji se koristi za vizualizaciju visokodimenzionalnih skupova podataka. On odabire dvije mjere sličnosti između parova točaka – jednu za visokodimenzionalne podatke, a drugu za dvodimenzionalnu ugradnju. Potom pokušava izgraditi dvodimenzionalno ugrađivanje koje minimizira divergenciju između vektora sličnosti između parova točaka u izvornom skupu podataka i sličnosti parova točaka u ugrađivanju [14].



**Slika 14. Primjer t-SNA [14]**

Na slici 14 (a) prikazan je visokodimenzionalni skup podataka, a na slikama (b) i (c) se primjenjuje t-SNE algoritam kako bi se smanjio broj dimenzija.

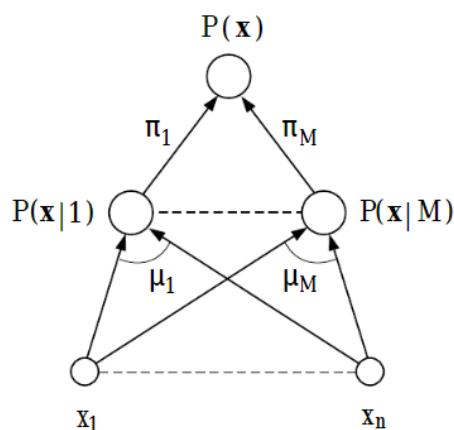
#### 4.2.3. *Strojno učenje pod nadzorom*

Ako za skup ulaza postoje označeni izlazi koji se unose u algoritam to se naziva učenje pod nadzorom. Cilj ovakvog tipa strojnog učenja je kreiranje modela koji je sposoban predvidjeti ispravan izlaz za nove ulazne podatke. Ukoliko postoje klasificirani povijesni podaci za neke od tipova kvara, moguće je pristupiti problemu na ovaj način. Neke tehnike učenja pod nadzorom su [11]:

##### 1. Neuronske mreže

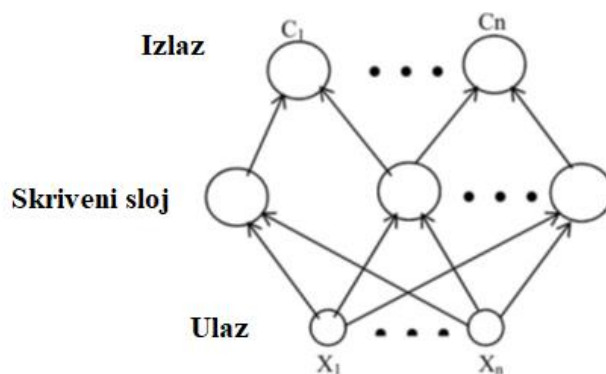
Mnogi pristupi PHM-a temelje se na umjetnim neuronskim mrežama. Neuronska mreža je graf temeljen na nekoliko međusobno povezanih brojčanih vrijednosti koje su pridodane u svaki čvor. Također ima skup ulaznih čvorova, izlaznih čvorova i skrivenih slojeva. Neuronske mreže naučene su kako bi skup podataka optimizirao parametre mreže i postigao se željeni izlaz. To je moguće ako se minimizira greška izlaza.

Neuronske mreže mogu se primijeniti u PHM-u kao statistički model i prediktivni algoritam pomoću procjene gustoće i predviđanja te klasifikacije i regresije. Za problem statističkog modeliranja i predviđanja, neuronska mreža može biti obučena za izradu statističkog modela koji se može koristiti za predviđanje izlaznih podataka za nove ulazne podatke. Procjena gustoće vjerojatnosti postiže se modeliranjem bezuvjetne distribucije podataka koji se obučavaju. Slika 15 prikazuje primjer neuronske mreže za problem gustoće vjerojatnosti koja se temelji na Gaussovoj distribuciji. Uz ulazne uzorke  $x$ , koji se nalaze na donjoj razini, veže se brojčana vrijednost. Oni su zatim povezani s određenim težinama ( $\mu_1, \mu_M$ ) pomoću kojih se računa Gaussova uvjetna vjerojatnost  $P(x|1)$  i  $P(x|M)$ . Te su vjerojatnosti ponderirane različitim proporcijama, a u izlaznom čvoru se dobiva suma.



**Slika 15. Neuronske mreže za problem gustoće vjerojatnosti temeljen na Gaussovoj distribuciji [11]**

Slika 16 prikazuje problem klasifikacije kada postoje više različitih ulaza  $X_n$ . Klasifikacija se postiže klasificiranjem ulaznih podataka u različite skupine temeljene na izlaznim podacima. U slučaju ulaznog vektora  $X$ , neuronska mreža klasificira ulazni vektor u jednu od klasa  $C_1, C_2, \dots, C_n$ . To je moguće primijeniti u PHM tako da izlazi mogu biti normalni rad sustava, anomalije ili kvarovi. Zatim se može koristiti regresija za ekstrapolaciju oštećenja ili kvara kako bi se procijenio RUL.

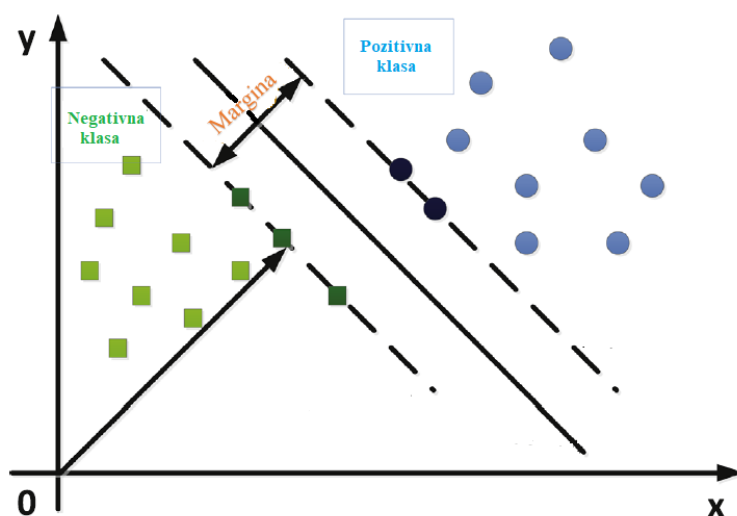


**Slika 16. Primjer neuronske mreže za problem klasifikacije [11]**

Jedna od glavnih prednosti neuronskih mreža je što je potreban samo mali broj podataka optimizirati za treniranje mreže i nisu potrebne prethodne pretpostavke o svojstvu podataka.

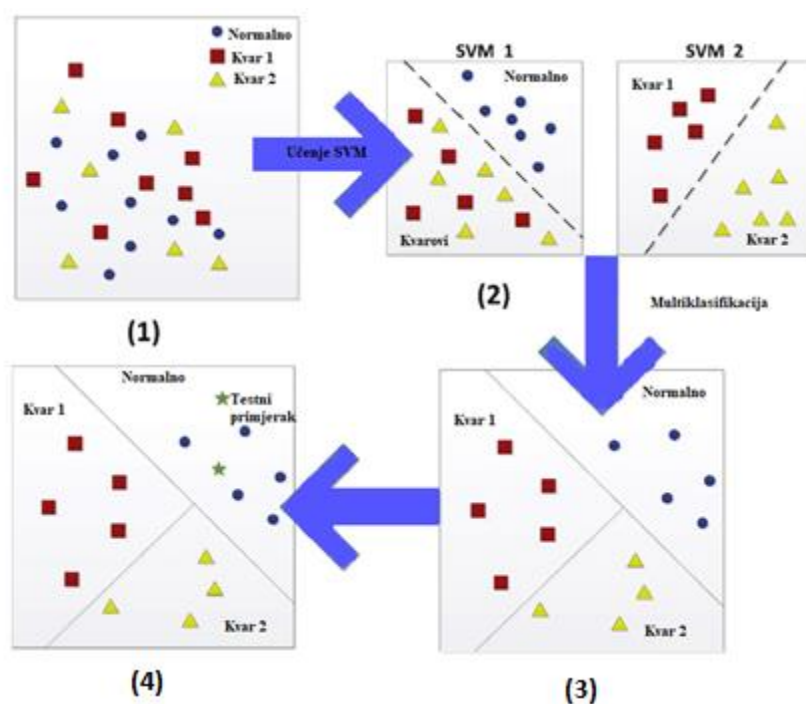
## 2. Metoda potpornih vektora

Metoda potpornih vektora (eng. *Support vector machine – SVM*) još jedan je pristup za rješavanje problema klasifikacije. Ideja je razdvajanje podataka za treniranje u dvije klase pomoću hiper-ravnine.



**Slika 17. Klasifikacije dvije klase korištenjem SVM-a [16]**

Slika 17 prikazuje podatke dvije klase, od kojih kvadrati prikazuju negativnu klasu, a krugovi prikazuju pozitivnu klasu. Metoda pomoćnih vektora pokušava staviti linearne granice u sredini te dvije klase. Maksimalna margina je prikazana kao udaljenost između dvije imaginarne linije. Točke, koje su smještene na imaginarnim linijama, nazivaju se pomoćnim vektorima. Pomoćni vektori su najznačajniji uzorci jer sadrže informacije o načinu stvaranja klasifikatora.



**Slika 18. Princip dijagnostike korištenjem SVM-a [16]**

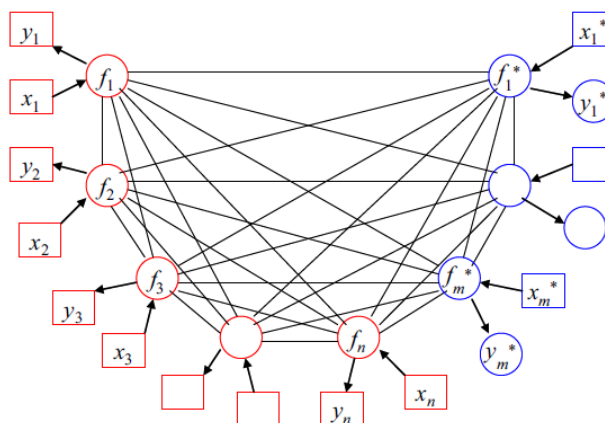
Dijagnostika kvarova u većini slučajeva je multiklasifikacija, to jest postoje više od dvije klase podataka (Slika 18). Nakon odabira podataka koji će služiti kao podaci za treniranje (1), primjenjuje se metoda pomoćnih vektora pomoću koje se dolazi do duple klasifikacije. Prvi SVM služi za klasifikaciju podataka u podatke koji prikazuju normalno stanje sustava i podatke koji predstavljaju kvarove (2). Sljedeći SVM predstavlja klasifikaciju različitih vrsta kvarova (2). U sljedećem koraku dolazi do kombinacije podataka (3), a u zadnjem koraku je moguće odrediti klasu testnog primjerka podatka [16].

Metoda pomoćnih vektora može se koristiti pri otkrivanju kvarova kod kugličnih ležajeva, induksijskih motora, pumpi, kompresora, ventila, turbina, i tako dalje.

### 3. Regresija Gaussovog procesa

Za određeni skup podataka postoji ulazni vektor  $x_i$  i skalarni izlaz ili cilj  $y_i$  (Slika 19). Ovakvi skupovi ulaznih i izlaznih parova nazivaju se primjerni ili eksperimentalni vrhovi. Zadatak regresije je, pomoću novog ulaznog podatka  $x_*$ , dobiti prediktivnu distribuciju promatrane vrijednosti  $y_*$  na temelju početnog skupa podataka.





**Slika 19. Grafički model za regresiju Gaussovog procesa [17]**

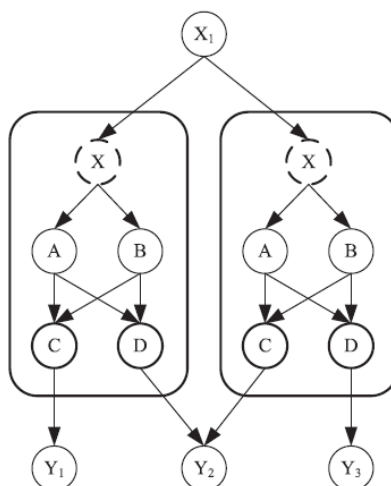
Gaussov proces je skup slučajnih varijabli, bilo kojih konačnih brojeva koji imaju zajedničku Gaussovu raspodjelu. Dok Gaussova distribucija prikazuje slučajne varijable. Gaussov proces opisuje raspodjelu preko funkcije [17].

Regresija Gaussovog procesa najčešće se koristi kod modeliranja nelinearnih sustava.

#### 4. Bayesova mreža (Bayesian networks – BN)

Bayesova mreža se sastoji od kvalitativnih i kvantitativnih dijelova. Kvalitativni dio je usmjereni aciklički graf u kojemu čvorovi predstavljaju sustavne varijable, a crte predstavljaju zavisnosti ili odnos uzroka i posljedica između varijabli.

Kvantitativni dio se sastoji od uvjetne tablice vjerojatnosti, koja predstavlja odnose između čvorova.

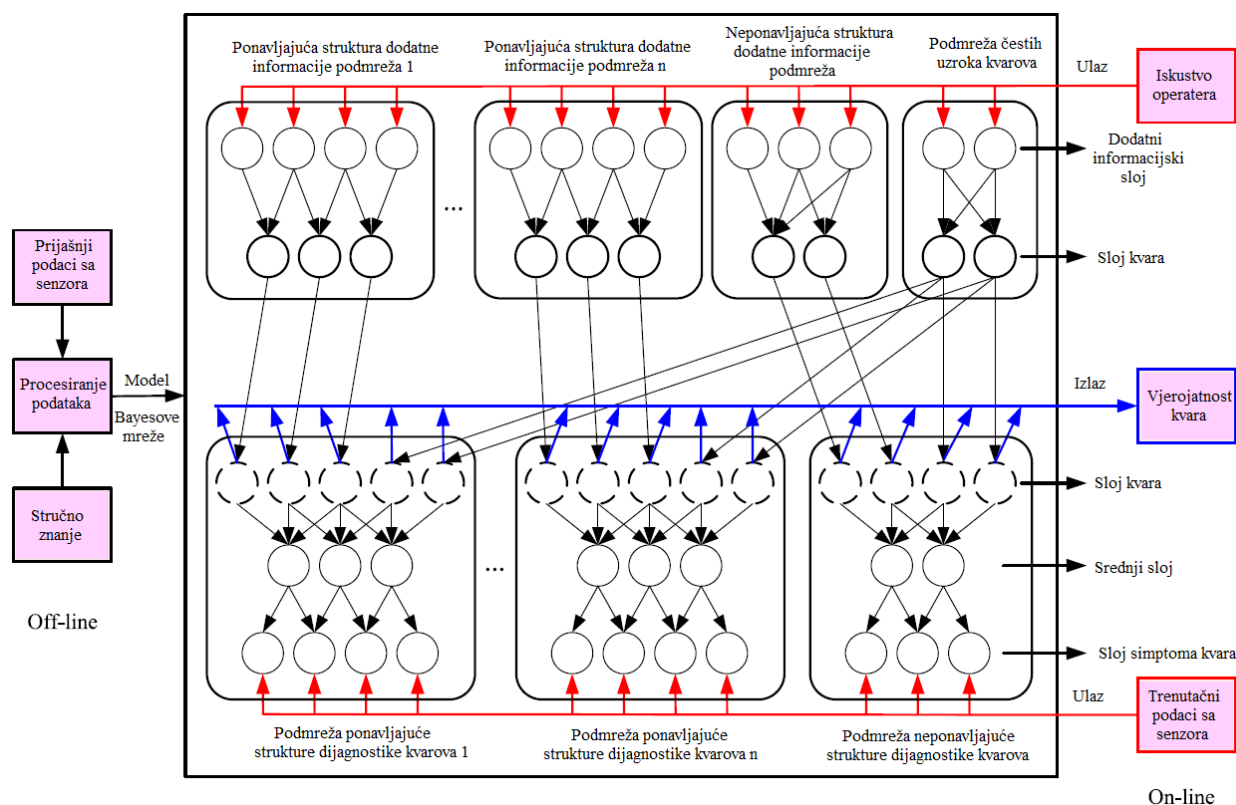


**Slika 20. Primjer Bayesove mreže [18]**

Kompleksniji Bayesov model se sastoji od sličnih dijelova. Te dijelove moguće je dobiti dijeljenjem Bayesove strukture nekoliko puta. Svaki takav generični dio mreže se naziva

klasa, a svaki dio dobiven dijeljenjem klase naziva se objekt. U primjeru na slici 20, ulazni čvor se dijeli na dva dijela. Ti dijelovi su povezani s ulaznim i izlaznim čvorovima. Ulazni atributi ( $X$ ) u opisu klase mogu se smatrati formalnim parametrima odgovarajuće funkcije, dok su stvarni parametri ( $X_1$ ) proslijeđeni na objekt identificirani kao nadređeni ulaznih atributa okolnom modelu.

Kompleksniji industrijski sustavi sadrže ponavljajuće strukture koje mogu biti kopije sličnih ili istih podsustava. Za takve sustave, razvija se dijagnostika kvarova u stvarnom vremenu korištenjem Bayesovih mreža. Metodologija se sastoji od dva dijela: *off-line* faza izrade Bayesove mreže i *on-line* faza dijagnostike kvarova. U *off-line* fazi, prijašnji podaci sa senzora, zajedno sa stručnim znanjem su prikupljeni kako bi se dobili kvarovi i njihovi simptomi. U *on-line* fazi, iskustvo operatera zajedno s trenutnim podacima sa senzora čine dodatne informacije i simptome kvarova kako bi se izvela dijagnostika kvarova [18]. Slika 21 prikazuje takvu metodologiju dijagnostike kvarova uz pomoć Bayesove mreže.



**Slika 21. Modeliranje metodologije dijagnostike kvarova na temelju Bayesove mreže [18]**

## 5. Skriveni Markovljev model

Skriveni Markovljev model (eng. *Hidden Markov model – HMM*) učinkovita je metoda prepoznavanja uzoraka koja se koristi u prepoznavanju govora, vizualnom

prepoznavanju i dijagnostici kvarova. Model se sastoji od dvije vrste stohastičkih varijabli, a to su skriveno stanje i vektor promatranja. Slijed skrivenog stanja je neprimjetan, ali se može procijeniti pomoću slijeda vektora promatranja [19].

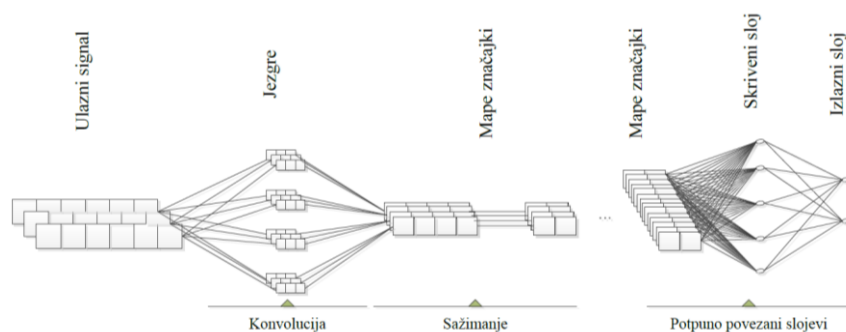
## 5. KONVOLUCIJSKA NEURONSKA MREŽA

Tehnologije strojnog učenja uvelike utječu na moderno društvo. Koriste za otkrivanje objekata na slikama, prepisivanje govora u tekst, otkrivanje proizvoda prema interesima korisnika i odabira relevantnih rezultata pretraživanja. Za ovakve primjene koristi se tehnika koja se naziva duboko strojno učenje. Metode dubokog strojnog učenja su reprezentativne metode učenja koje se sastoje od nekoliko razina, dobivenih jednostavnim sastavljanjem nelinearnih modela. Svaki takav model transformira ulazne složenije podatke te dolaze do apstraktne razine. Uz takvu dekompoziciju, moguće je naučiti kompleksne funkcije [20]. Duboko strojno učenje upotrebljava duboke neuronske mreže, koje se od običnih razlikuju po broju skrivenih slojeva. Duboke neuronske mreže se sastoje od dva ili više skrivena slojeva, dok obične mogu imati jedna ili dva sloja. Pokazalo se da su dobre u otkrivanju unutarnje strukture visokodimenzionalnih podataka te se iz tog razloga koriste u znanosti, ekonomiji i upravi.

Konvolucijska neuronska mreža je duboka neuronska mreža koja koristi konvoluciju na mjestu množenja matrice u barem jednom od svojih slojeva. Izgrađena je da prepozna vizualne uzorke sa slika s minimalnim procesiranjem. Mreža koristi umjetne neurone koji su funkcija koja od ulaznog signala stvara izlazni. Broj neurona ovisi o zadatku koji je potrebno riješiti, te postoji više različitih načina spajanja neurona.

Konvolucijsku neuronsku mrežu čine sljedeći slojevi (slika 22):

- Ulazni sloj
- Konvolucijski sloj
- ReLU
- Sloj sažimanja
- Potpuno povezani sloj
- Izlazni sloj



Slika 22. Opća struktura konvolucijskih neuronskih mreža [21]

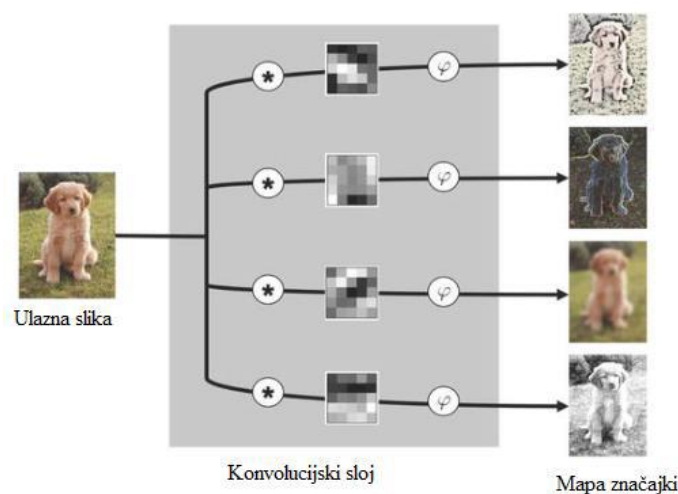
## 5.1. Konvolucijski sloj

Ulazni signal, koji prethodi konvolucijskom sloju su slike ili signali prikupljeni sa senzora. Takvi signali imaju tri dimenzije: visinu, širinu i dubinu. Visina i širina ulaznog signala ovise o visini i širini slike, dok dubina može biti različita. Ako je slika crno-bijela ili jednobojna dubina je jednaka broju 1, a ukoliko je u boji ona iznosi 3. To ovisi o broju osnovnih boja (crvena, zelena, plava) na slici.

Konvolucijski sloj je glavni element konvolucijskih neuronskih mreža u kojemu se odvija matematička operacija, konvolucija. Konvolucija je integriranje dviju funkcija po intervalu, gdje jedna funkcija utječe na drugu funkciju kako bi se dobila treća funkcija.

$$(f * g)(t) = \int_0^t f(t - \tau)g(\tau)d\tau \quad 5.1$$

Konvolucijska neuronska mreža spaja svaki neuron konvolucijskog sloja samo s regijom povezivanja ulaznog sloja koja se naziva lokalna receptivna regija ili polje. Njezina veličina ovisi o veličini jezgre (eng. *kernel*) koja zajedno s brojem jezgri i korakom čini hiperparametre koje je potrebno odrediti prije učenja neuronske mreže. Operacija konvolucije se odvija nad svakim lokalni receptivnim poljem, gdje se svako polje množi s prikladnom jezgrom. Produkti koji nastaju se zbrajaju u polje u mapi značajki. Za konvoluciju je potrebno odabrati veličinu i broj jezgri koje ulaze preko ulazne matrice i obavljaju operaciju konvolucije. Rezultat toga je mapa značajki koja predstavlja novu matricu za obrađivanje u novom sloju. Mape značajki naglašavaju značajke originalne slike. Slika 23 prikazuje procese u konvolucijskom sloju u kojemu prvi krug predstavlja konvoluciju, a drugi krug predstavlja aktivacijsku funkciju. Kvadrati, između tih operacija, predstavljaju konvolucijske jezgre. U ovom slučaju se konvolucijski sloj sastoji od četiri jezgre, pomoću kojih će nastati četiri mape značajki.

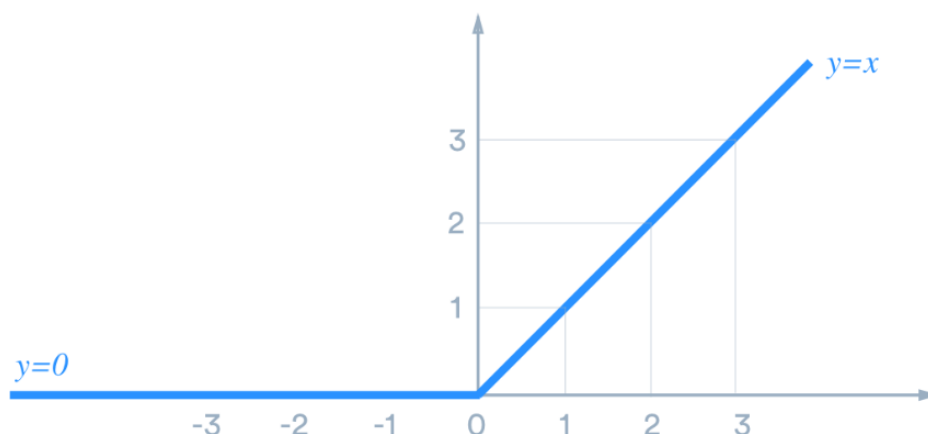


Slika 23. Proces konvolucije i nastajanje mape značajki [22]

Jezgre, u konvolucijskim slojevima, su dvodimenzionalne matrice. Na slici 23 veličina jezgri je  $5 \times 5$ . Kako je već spomenuto prije, vrijednost matrice jezgri određuje se tijekom procesa treniranja, stoga se ove vrijednosti kontinuirano uče tijekom cijelog procesa učenja mreže.

Nakon postupka konvolucije, potrebno je dobivene mape značajki provesti kroz nelinearne aktivacijske funkcije kako bi se modelirale nelinearnosti [22]. Iako postoji nekoliko aktivacijskih funkcija, u konvolucijskim neuronskim mrežama se koristi ReLU aktivacijska funkcija. Ona je definira jednadžbom 5.2 i prikazana je na slici 24.

$$f(x) = \max(0, x) \quad 5.2$$

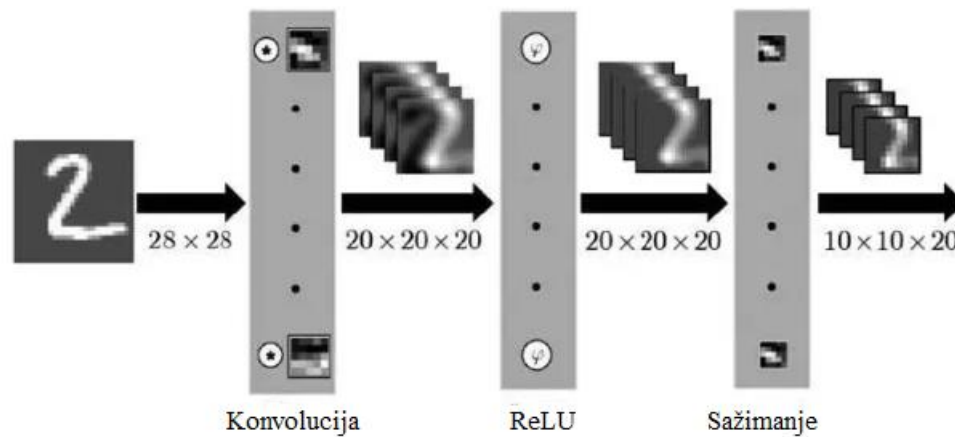


**Slika 24. ReLU aktivacijska funkcija**

ReLU aktivacijska funkcija je pogodna za korištenje u dubokim neuronskim mrežama jer osigurava konstantnu vrijednost gradijenta (0 ili 1), s obzirom da je rezultat uvijek različit od nule. Prilikom povratnog prostiranja gradijent se ne smanjuje u dubljim slojevima, što rezultira bržim učenjem. Ukoliko je negativna vrijednost neurona, deriviranjem aktivacijske funkcije dobiva se nula, što u praktičnom slučaju znači gašenje neurona mreže. Dodatni sloj za normalizaciju podataka (eng. *batch normalization layer*) smanjuje vjerojatnost pojave većeg broja gašenja neurona [21].

## 5.2. Sloj sažimanja

Sažimanje (eng. *pooling*) je postupak smanjenja veličine slike kombiniranjem susjednih piksela određenog područja slike u jednu reprezentativnu vrijednost [22]. Unutar konvolucijske neuronske mreže, sažimanje se koristi s ciljem smanjenja dimenzije ulaza i povećanja neosjetljivosti mreže kroz uzrokovanje. Pritom se ne mijenja broj mapa značajki, već samo njihove dimenzije (Slika 25) [21].



**Slika 25. Obrada slike kroz konvoluciju, ReLU i sažimanje [22]**

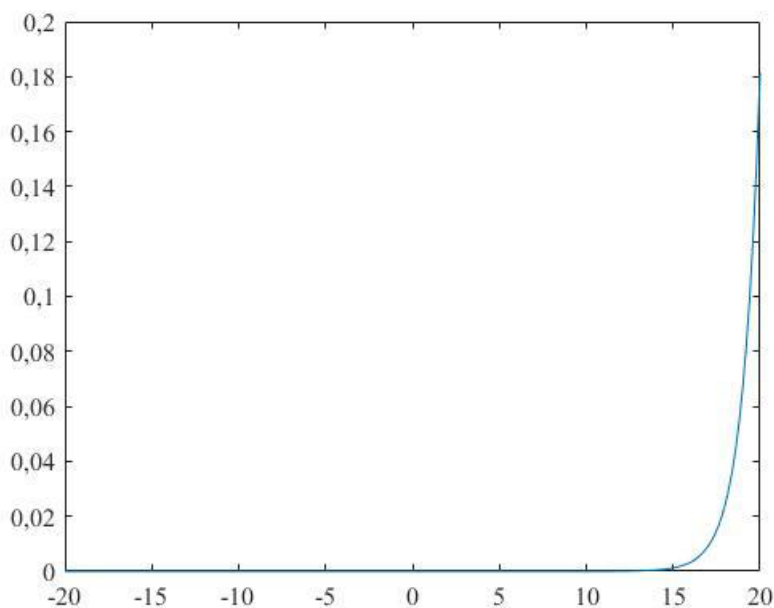
S obzirom da se sažimanjem smanjuje dimenzionalost ulaza, također se smanjuje broj parametara za učenje. Također se smanjuje osjetljivost konvolucijske mreže na različitost ulaza. To je korisno kod promjene položaja značajke koja se nastoji otkriti pomoću jezgre u konvolucijskom sloju unutar promatranog lokalnog receptivnog polja [21].

### 5.3. Izlazni sloj

Softmax funkcija je implementirana u neuronsku mrežu prije izlaznog sloja odnosno klasifikacije. Matematički gledano ona generalizira sigmatoidnu funkciju, koja se koristi kao izlazni neuron za klasifikatore. Ima sposobnost transformacije  $k$ -dimenzionalnog vektora  $x$  sastavljenog od neskaliiranih realnih vrijednosti u  $k$ -dimenzionalni vektor  $f(x)$  čije komponente sadrže vrijednosti u intervalu  $(0,1)$  te im je zbroj jednak 1, kako je prikazano na jednadžbi (5.3) i na slici 26 [21].

$$f(x)_j = \frac{e^{x_j}}{\sum_{i=1}^k e^{x_i}}, j \in \{1, 2, \dots, k\},$$

$$\sum_{j=1}^k f(x)_j = 1 \quad 5.3$$



**Slika 26. Softmax funkcija**

Izlaz konvolucijske neuronske mreže predstavlja stupanj vjerojatnosti mreže na donesenu odluku. Što je veći broj to je vjerojatnost odluke točnija.

#### 5.4. Određivanje hiperparametara modela

Svaki model strojnog učenja sadrži parametre, koji reguliraju učenje. Takvi parametri podešavaju se prije samog procesa učenja te uvelike mogu promijeniti ishod postupka učenja. Oni su različiti od parametara koje model uči i podešava tijekom treninga i nazivaju se hiperparametrima [21]. Shodno tome postoje dvije vrste parametara u neuronskim mrežama:

- parametri koje mreža uči i podešava tijekom treninga i
- hiperparametri koji se zadaju prije samog procesa učenja, te se ne uče naknadno.

Hiperparametri se mogu podijeliti u tri kategorije [21]:

1. hiperparametri s kontinuiranim realnim brojevima koji mogu poprimiti bilo koju vrijednost unutar određenog raspona, a primjeri su: koeficijent učenja, momentum, parametar regularizacije;
2. hiperparametri s diskretnim cjelobrojnim vrijednostima poput broj konvolucijskih slojeva, broj jezgri u svakom sloju, veličina jezgri, broj epoha i drugi te
3. hiperparametri koji su diskretne i/ili binarne vrijednosti i sadrže konačan broj mogućnosti: odabir optimizatora, odabir aktivacijske funkcije, metoda smanjivanja koeficijenata učenja.



Različiti skupovi podataka posljedično znače i različite optimalne hiperparametre.

S obzirom da postoji velika količina hiperparametara i veliki rasponi istih, koji utječu na ishode učenja, potrebno je izabrati optimalne kako bi se dobila najveća uspješnost klasifikacije, odnosno najbolji ishod modela. Također treba uzeti u obzir vrijeme potrebno za treniranje mreže, tako da nije moguće isprobavati različite kombinacije hiperparametara u svrhu dobivanja optimalnih rezultata.

Postojeći hiperparametri konvolucijske neuronske mreže su sljedeći [21]:

- *dubina mreže i broj konvolucijskih slojeva*

- *broj jezgri*

Svaki konvolucijski sloj sadrži otprije odabrani broj jezgri. Broj jezgri utječe na broj mapi značajka (jednak je broj jezgri i mapi značajka) u svakom sloju, a težine se u njima mijenjaju tijekom treniranja mreže.;

- *veličina jezgre*

Dimenzije jezgre se definiraju širinom i visinom matrice. Jezgre su kvadratne matrice, dimenzija 3x3 ili 5x5. Svako polje unutar jezgre predstavlja jednu težinu, a na primjer jezgra dimenzija 3x3 će imati 9 težina. U slučaju da se izabere premala veličina jezgre, mreža možda neće biti sposobna prenijeti sve bitne informacije u procesu konvolucije, dok prevelika povećava vrijeme učenja.;

- *dopunjavanje*

Dimenzije ulaznih matrica mogu se povećati dodajući nule oko rubova kako bi se dobila mapa značajki jednakih dimenzija kao i ulazna matrica. Kod ovog se hiperparametra odlučuje hoće li se dopunjavanje koristiti ili neće.;

- *korak s*

Broj za koji se je potrebno pomaknuti u vertikalnom i horizontalnom smjeru kod operacija konvolucije i sažimanja. U slučaju korištenja 1D signala, pomicanje se vrši samo u horizontalnom smjeru.;

- *redoslijed i broj slojeva*

Dokazano je da mreže s više slojeva zahtijevaju mnogo manji broj neurona da bi se postigli isti rezultati i bolje generaliziraju na temelju nepoznatih podataka.;

- *koeficijent učenja*

Odabire se u rasponu između 0,0001 i 0,1, a smatra se jednim od ključnim hiperparametra, jer o njemu ovisi konvergencija mreže.;

- *aktivacijska funkcija*

- *broj epoha*

Epohu sačinjava jedan prolaz svih uzoraka kroz neuronsku mrežu za vrijeme treninga. Nakon svake epohe se može vidjeti pogreška koju mreža daje, odnosno moguće je vidjeti uspješnost klasificiranih podataka. Manjim brojem epoha se skraćuje vrijeme treniranja, ali i pogoršava istreniranost mreže.;

- *veličina mini grupe podataka*

O ovom hiperparametru ovisi koliko će uzoraka u jednom prolazu proći kroz mrežu. Dobar izbor veličine mini grupe znatno ubrzava učenje mreže, ne gubeći previše na njenoj točnosti. Ukoliko se koristi algoritam gradijentni spust s mini grupama, težine će se ažurirati nakon svake mini grupe.

- *parametri regularizacije*

Odabir iznosa  $\lambda$  ukoliko se radi o regularizaciji L1 ili L2 vektorskim normama. Osim L1 i L2 regularizacije, može se koristiti i tehnika isključivanja (engl. dropout) neurona tijekom učenja, no generalno se kod konvolucijskih neuronskih mreža češće koristi tehnika normalizacije grupe (eng. *batch normalization*), koja se pokazuje efektivnijom.;

- *optimizacijska tehnika*

- *funkcija gubitka*

Ovisno o problemu (klasifikacija ili regresija) i ulaznim podacima odabire se funkcija gubitka. Unakrsna provjera se najčešće koristi za klasifikaciju, a za regresiju se koriste: srednja kvadratna pogreška, korijen srednje vrijednosti kvadrata pogreške, srednja apsolutna pogreška, i dr., te

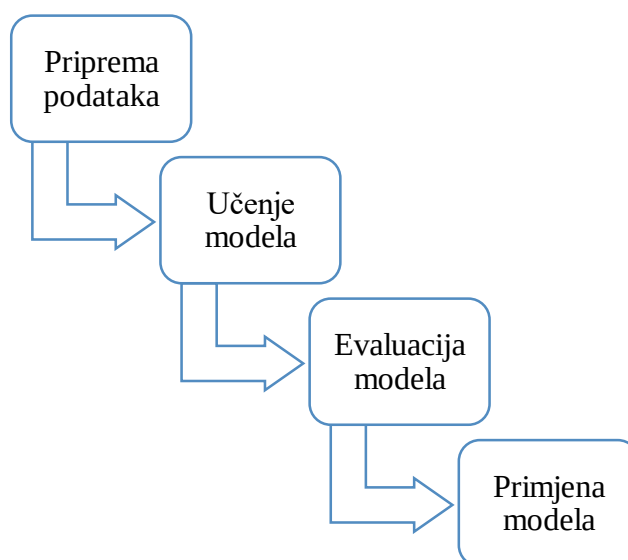
- *sloj sažimanja (eng. pooling).*

## 6. RAZVOJ KORISNIČKOG SUČELJA

Trendovi razvoja računalnih sustava sa sposobnošću predviđanja stanja opreme kreću se u smjeru prepoznavanja anomalije u radu opreme te predlaganja aktivnosti održavanja kojima će se osigurati normalna funkcionalnost sustava. To je moguće postići uz metode navedene u 4. poglavlju, a najveću primjenu su pronašle metode strojnog učenja pod nadzorom i bez nadzora. Kako je navedeno u 5. poglavlju, konvolucijska neuronska mreža je metoda strojnog učenja pod nadzorom koja je pronašla široku primjenu u područjima poput dijagnostike kvarova ležajeva, motora, procjena preostalog korisnog vijeka sustava, izdvajanje značajki u detektiranju oštećenja, otkrivanje kvarova na temelju vibracija i mnogim drugim.

U praktičnom dijelu rada bit će korišteni označeni podaci prikupljeni simuliranjem različitih kvarova na eksperimentalnom simulacijskom postavu. Takvi označeni izlazni podaci (kvarovi), zajedno s ulaznim podacima, su karakteristika strojnog učenja pod nadzorom. Te je upravo to razlog zbog kojeg će se koristiti algoritam konvolucijske neuronske mreže u praktičnom dijelu rada.

Kako je navedeno u uvodnom poglavlju, trenutačno ne postoji jedinstveno korisničko sučelje koje objedinjuje modele za prikupljanje podataka, učenje i evaluaciju neuronske mreže, tumačenje podataka te jednostavnu integraciju pojedinih algoritama za strojno učenje. Cilj praktičnoga rada je stvoriti upravo takvo korisničko sučelje u razvojnoj okolini MATLAB čiji su glavni zadaci prikazani na slici 27.



Slika 27. Glavni zadaci razvijenog korisničkog sučelja

Radi mogućeg razumijevanja i korištenja korisničkog sučelja u sljedećim potpoglavljima detaljno će se objasniti tijekom izrade te svi zadaci koje je moguće izvršiti u razvijenom korisničkom sučelju.

### **6.1. Razvojna okolina - MATLAB Application Designer**

MATLAB je razvojna okolina koja služi za računanje, vizualizaciju i programiranje, u kojemu se problemi i rješenja iskazuju u uobičajenoj matematičkoj notaciji. Tipične primjene MATLAB-a su numerička i simbolička izračunavanja, razvoj algoritama, modeliranje i simulacija sustava, eksperimentiranje sa sustavima u stvarnome svijetu, obrada i vizualizacija podataka, znanstveno-inženjerska grafika te razvoj aplikacija, uključujući i razvoj grafičkih korisničkih sučelja. Osim osnovnog programskog okruženja, MATLAB se sastoji i od velikog broja dodatnih funkcija koji su organizirani prema području primjene (paketi za obradu signala, upravljanje sustavima, neuronske mreže, statistiku i slično) [23].

Jedna od mogućnosti MATLAB-a je razvoj korisničkog sučelja. Razvojna okolina za stvaranje korisničkog sučelja sastoji se od dva modula. Prvi čine različite vizualne komponente koje se prikazuju u korisničkom sučelju. One se na jednostavan način mogu postaviti na unutar budućeg korisničkog sučelja te je uz pomoć određenih naredbi moguće odrediti njihov točan raspored. Neke od mogućih komponentata su tipke, izbornici, numerička polja, tablice, treptajući elementi i mnogi drugi. Drugi dio je programiranje ponašanja korisničkog sučelja odnosno povezivanja naredbi s vizualnim komponentama. Ova razvojna okolina sastoji se od potpuno integrirane verzije MATLAB-a što omogućuje izvršavanje svih funkcija koje sama razvojna okolina podržava. Razvojna okolina omogućuje automatsku provjeru napisanih funkcija te javlja bilo kakva upozorenja i pogreške, prema kojima je zatim moguće izmijeniti napisane funkcije. Nakon razvijenog korisničkog sučelja, moguće je isti podijeliti s drugim korisnicima MATLAB-a preko internetske stranice MATLAB-a. Također je moguće stvoriti web aplikaciju kojoj se pristupa direktno preko internetske tražilice bez dodatnog instaliranja razvojne okoline MATLAB-a [24].

### **6.2. Struktura i učitavanje ulaznih podataka**

Dobiveni podaci sa senzora spremljeni su u formatu čije su vrijednosti odvojene sa zarezom (eng. *Comma seperated value* - CSV). Podaci su prikupljeni u *Laboratoriju za održavanje Fakulteta strojarstva i brodogradnje* uz pomoć mjerne opreme koju su činili optički senzor i senzor akceleracije. Oni su za potrebe eksperimenta montirani na simulator kvarova rotacijske opreme [21].

Kako bi se ti podaci mogli obrađivati potrebno ih je pretvoriti u format koji koristi razvojna okolina MATLAB. Takav format naziva se MAT i se sastoji od podataka poput varijabli, funkcija, redova i drugih informacija. Razlog pretvaranja je kreiranje skladišta podataka (eng. *datastore*), koje služi za sakupljanje podataka koji su preveliki da bi se pohranili u memoriju. Ono omogućuje čitanje i obradu podataka pohranjenih u više datoteka na disku, udaljenom mjestu ili bazi podataka kao jedinstveni entitet [24].

Podaci prikupljeni sa senzora koji su spremljeni u CSV datoteke te se koriste u ovom slučaju, su broj okretaja, broj mjerenja, tip stanja stroja, te podaci u ravninama *X*, *Y*, *Z* i *Z1* (Slika 28). Postoji 16 datoteka koje su razvrstane prema tipu stanja stroja i broju okretaja. Prema tipu stanja stroja postoji osam skupina:

- normalno stanje (eng. *Normal state - NS*)
- neravnoteža motora (eng. *Imbalance rotor fault - IMRF*)
- ekscentrični rotor (eng. *Eccentric rotor fault ERF*)
- nagnuti rotor (eng. *Cocked rotor fault - CRF*)
- oštećenje vanjske staze kotrljanja kotrljajućeg ležaja (eng. *Outer race bearing fault - ORBF*)
- oštećenje unutarnje staze kotrljanja kotrljajućeg ležaja (eng. *Inner race bearing fault - IRBF*)
- oštećenje kotrljajućeg elementa kotrljajućeg ležaja (eng. *Ball bearing fault - BBF*)
- kombinirani kvar ležaja (eng. *Combinated bearing fault - CBF*)

Prema broju okretaja postoje dvije skupine:

1. 1000 okretaja
2. 1500 okretaja.

| Acquired Data X | Broj okretaja | Broj mjerenja | Acquired Data Y | Acquired Data Z | Acquired Data Z1 | Tip kvara |
|-----------------|---------------|---------------|-----------------|-----------------|------------------|-----------|
| -0.003131       | 1000.0        | 1.0           | -0.006377       | 0.012013        | -0.045504        | BBF       |
| -0.022757       | 1000.0        | 1.0           | -0.030058       | 0.064368        | -0.266319        | BBF       |
| -0.071513       | 1000.0        | 1.0           | -0.06141        | 0.163218        | -0.723638        | BBF       |
| -0.12995        | 1000.0        | 1.0           | -0.067929       | 0.26823         | -1.248.464       | BBF       |
| -0.157251       | 1000.0        | 1.0           | -0.029795       | 0.317496        | -1.580.812       | BBF       |
| -0.149168       | 1000.0        | 1.0           | 0.033544        | 0.265584        | -157.806         | BBF       |
| -0.153393       | 1000.0        | 1.0           | 0.078097        | 0.122448        | -1.237.506       | BBF       |
| -0.214093       | 1000.0        | 1.0           | 0.088115        | -0.055151       | -0.661822        | BBF       |
| -0.304648       | 1000.0        | 1.0           | 0.091304        | -0.216255       | -0.042079        | BBF       |
| -0.336351       | 1000.0        | 1.0           | 0.100055        | -0.320782       | 0.397065         | BBF       |
| -0.262147       | 1000.0        | 1.0           | 0.087393        | -0.327806       | 0.540762         | BBF       |
| -0.14168        | 1000.0        | 1.0           | 0.044151        | -0.240545       | 0.530542         | BBF       |
| -0.075993       | 1000.0        | 1.0           | 0.005435        | -0.127097       | 0.598214         | BBF       |
| -0.097379       | 1000.0        | 1.0           | 0.004792        | -0.059985       | 0.754297         | BBF       |
| -0.135532       | 1000.0        | 1.0           | 0.032378        | -0.065803       | 0.860294         | BBF       |
| -0.100057       | 1000.0        | 1.0           | 0.049053        | -0.124973       | 0.859902         | BBF       |
| 0.018259        | 1000.0        | 1.0           | 0.036916        | -0.188342       | 0.823199         | BBF       |
| 0.141402        | 1000.0        | 1.0           | 0.021786        | -0.21836        | 0.880274         | BBF       |
| 0.172855        | 1000.0        | 1.0           | 0.034418        | -0.219271       | 1.022.605        | BBF       |
| 0.088273        | 1000.0        | 1.0           | 0.056822        | -0.200792       | 107.329          | BBF       |
| -0.026005       | 1000.0        | 1.0           | 0.046232        | -0.144228       | 0.975577         | BBF       |
| -0.067379       | 1000.0        | 1.0           | 0.009401        | -0.04436        | 0.830317         | BBF       |
| -0.035272       | 1000.0        | 1.0           | -0.006731       | 0.056592        | 0.66839          | BBF       |
| -0.013642       | 1000.0        | 1.0           | 0.013071        | 0.123718        | 0.428075         | BBF       |
| -0.061497       | 1000.0        | 1.0           | 0.037866        | 0.172029        | 0.077875         | BBF       |
| -0.141545       | 1000.0        | 1.0           | 0.03746         | 0.220884        | -0.32247         | BBF       |
| -0.164958       | 1000.0        | 1.0           | 0.019573        | 0.266833        | -0.669231        | BBF       |
| -0.108436       | 1000.0        | 1.0           | 0.01707         | 0.300796        | -0.899245        | BBF       |
| -0.039014       | 1000.0        | 1.0           | 0.044664        | 0.28878         | -0.960076        | BBF       |

**Slika 28. Prikaz prikupljenih podataka koji su spremljeni u CSV datoteku**

Od navedenih osam stanja, prvo je normalno stanje simulirane opreme, a preostala označavaju simulirane kvarove pomoću modula na simulatoru. Prikupljeni i označeni podaci različitih stanja opreme definirani su kao klase strojnog učenja pod nadzorom. Količine prikupljenih podataka ovise o pojedinom problemu za koji se želi naučiti neuronska mreža. S obzirom na mogućnosti razvijenih hardvera, u današnje vrijeme moguće je prikupiti velike količine podataka. Nadalje, modeli dubokog strojnog učenja za učenje zahtijevaju velik broj podataka. U konkretnom slučaju, 220 puta je ponovljeno mjerenje za svih šesnaest kombinacija. Svakim mjerenjem se za svaki ispitni uzorak prikupio 25 600 podataka, što je ukupno 1,87 GB, koji trebaju biti obrađeni tehnikom dubokog strojnog učenja.

Prvi korak je učitavanje CSV datoteke, poput datoteke prikazane na slici 28, koja se želi pretvoriti u MAT format. To je moguće napraviti uz pomoć funkcije koja obrađuje velike količine podataka te funkcije koja će preoblikovati CSV datoteke.

Slika 29 prikazuje pozadinski kod koji će se pokrenuti nakon pritiska na tipku u korisničkom sučelju koji je točno previđen za učitavanje CSV datoteke i pretvaranje u MAT format. Postoji 16 datoteka koje je potrebno pretvoriti, te će se svaka od njih pretvoriti pojedinačno.

```
% Value changed function: Ucitaj
function UcitajValueChanged(app, event)
    datalocation=uigetdir('D:\Diplomski\Lusetic\');
    app.delimiter=';';
    app.RowDelimiter='\n';
    app.ds = tabularTextDatastore(datalocation,'FileExtensions','.csv','IncludeSubfolders',true,'Delimiter',app.delimiter, 'RowDelimiter', app.RowDelimiter, 'SelectedVariableNames', {'Tipkvara', 'BrojMjerenja', 'AcquiredDataX', 'AcquiredDataY', 'AcquiredDataZ'});
    app.CR1500rpmtme=readall(app.ds);

    % Kreiraj file-ove za svako mjerenje
    % Normal state
    app.sample_rate =25600; % ovo je broj zapisa
    app.cr_X_os = app.CR1500rpmtme(:,3);
    app.cr_Y_os = app.CR1500rpmtme(:,4);
    app.cr_Z_os = app.CR1500rpmtme(:,5);
    app.cr_num_examples = unique(app.CR1500rpmtme(:,2));

    % Broj_mjerenja = 1:3000;
    app.vib_data_cr = reshape([app.cr_X_os app.cr_Y_os app.cr_Z_os], app.sample_rate, length(app.cr_num_examples), 3);
end
```

**Slika 29. Primjer pozadinskog koda za pretvaranje CSV datoteke u MAT format**

Drugi korak je spremanje pretvorene datoteke u određenu mapu koju je moguće izabrati te određivanje imena datoteka. S jednom CSV datotekom u ovom konkretnom primjeru nastaje 220 datoteka MAT formata određenog stanja sustava i određenog broja okretaja koji se označavaju u imenu datoteke. Za daljnje korištenje potrebno je definirati naziv mapa (Slika 30), u koje će biti smještene prethodno pretvorene datoteke.



**Slika 30. Mape s 220 MAT datoteka**

### 6.3. Podjela podataka za učenje, testiranje i validaciju

Sljedeći korak povezan je s učenjem konvolucijske neuronske mreže. Nakon što su se podaci pretvorili u MAT format te ih je moguće učitati u MATLAB i uz njihovu pomoću provesti učenje mreže. Podaci se dijele na tri dijela:

#### 1. Podaci za učenje mreže

Oni se koriste kod učenja mreže zajedno s određenim slojevima i opcijama koje je potrebno zadati kako bi se pokrenulo učenje (npr. veličina mini serije, momentum, koeficijent učenja, prikaz grafa učenja, podaci za validaciju itd.)

## 2. Podaci za testiranje mreže

Oni se koriste kod izračunavanja uspješnosti klasifikacije modela gdje se stvarna klasa testiranih podataka uspoređuje s predviđenom klasom testiranih podataka.

## 3. Podaci za validaciju mreže.

Koriste se prilikom učenja mreže, a potrebno ih je odrediti u opcijama učenja mreže. Oni služe kako bi se već prilikom učenja mogla izračunati uspješnost klasifikacije modela u pojedinim iteracijama učenja.

#### 6.4. Određivanje hiperparametara za učenje mreže

Prije početka treniranja mreže, potrebno je odrediti hiperparametre na temelju kojih će se provoditi treniranje. Za ovaj primjer koristiti će se šest hiperparametara (Poglavlje 5.4) te su za njih definirane i sljedeće granice.

**Tablica 4. Rasponi hiperparametra [21]**

| Naziv hiperparametra                  | Raspon       | Tip podataka                        |
|---------------------------------------|--------------|-------------------------------------|
| <b>Broj naslaganih blokova/dubina</b> | 0 - 8        | Cjelobrojni                         |
| <b>Veličina jezgri</b>                | 4 - 64       | Cjelobrojni                         |
| <b>Broj jezgri</b>                    | 4 - 24       | Cjelobrojni                         |
| <b>Faktor skaliranja</b>              | 1 - 3        | Cjelobrojni                         |
| <b>Koeficijent učenja</b>             | 0,001 – 0,01 | Kontinuirani,<br>logaritamska skala |
| <b>Momentum</b>                       | 0,80 – 0,95  | Kontinuirani,<br>logaritamska skala |

S obzirom na velike raspone hiperparametra te njihove kombinacije, moguće je dobiti različite uspješnosti klasifikacije konvolucijskih neuronskih mreža. Zbog dugotrajnog postupka izračunavanja optimalnih hiperparametara, koristit će se već prije izračunate vrijednosti hiperparametara. [21]. Sljedeća tablica prikazuje vrijednosti hiperparametara s kojima je dobivena uspješnost klasifikacije mreže od 99,537 %.

**Tablica 5. Vrijednosti hiperparametra [21]**

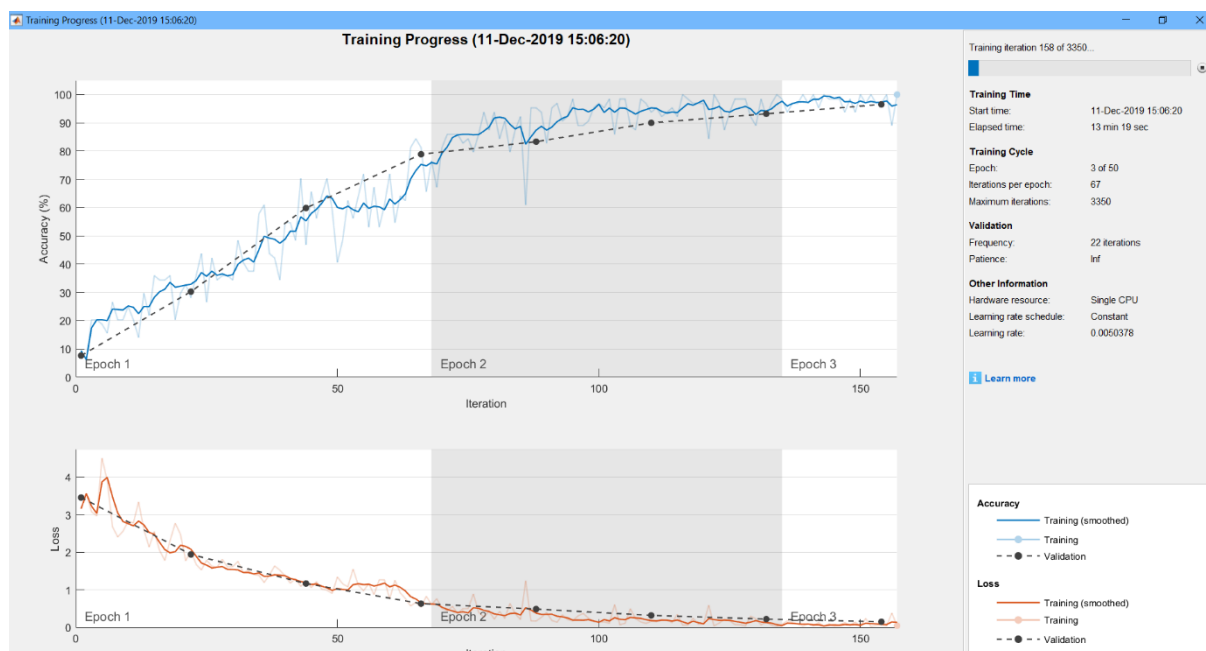
| Hiperparametar                        | Vrijednost varijable |
|---------------------------------------|----------------------|
| <b>Broj naslaganih blokova/dubina</b> | 1                    |



|                           |             |
|---------------------------|-------------|
| <b>Veličina jezgri</b>    | 51          |
| <b>Broj jezgri</b>        | 6           |
| <b>Faktor skaliranja</b>  | 1           |
| <b>Koeficijent učenja</b> | 0,005037766 |
| <b>Momentum</b>           | 0,872358819 |

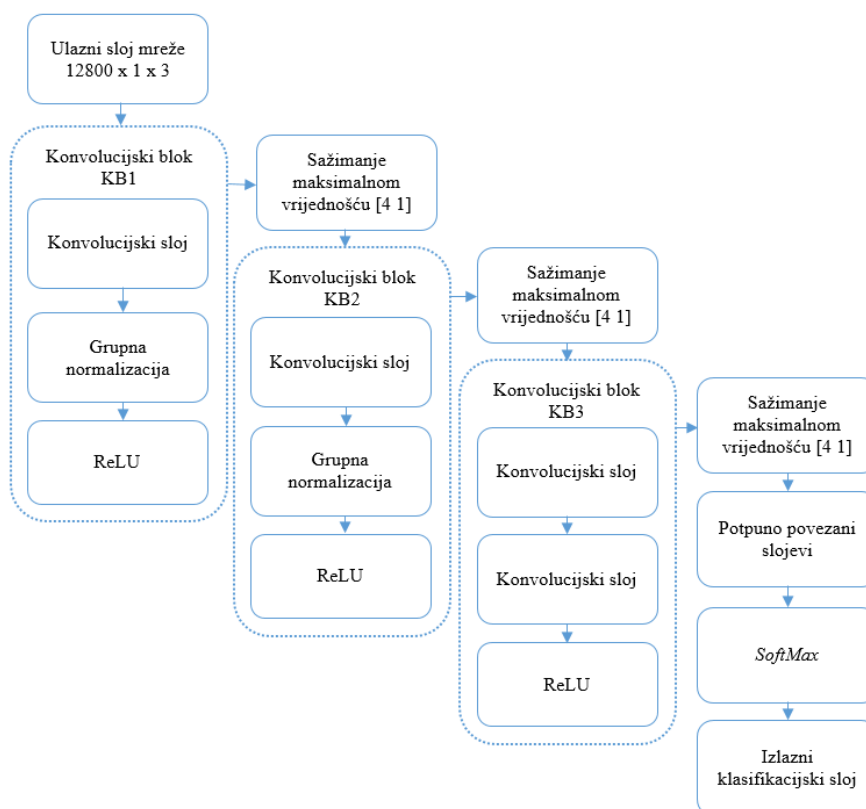
### 6.5. Integracija algoritma za učenje

U MATLAB-u postoji opcija praćenja učenja neuronske mreže (slika 31). Gornji graf na slici 31 prikazuje rast odnosno pad uspješnosti klasifikacije prilikom učenja neuronske mreže kroz određene epohe i iteracije. U ovom slučaju broj iteracija kroz jednu epohu iznosi 67, a on ovisi o broju podataka koji se koriste za treniranje mreže. Što je veći broj podataka, to je veći broj iteracija. Maksimalan broj epoha određen je na 50, pri čemu se jedna epoha učenja smatra završenom u trenutku kad svi podaci iz skupa za učenje prođu kroz strukturu algoritma za učenje. Tri linije na grafu prikazuju tri različite uspješnosti klasifikacije: prva uspješnost je uspješnost klasifikacije za svaku mini seriju (eng. *mini batch*), druga linija prikazuje uspješnost klasifikacije nakon primjene algoritma za izgladivanje i treća (crna linija) prikazuje uspješnost klasifikacije s obzirom na validacijski skup podataka (Poglavlje 6.3.). Donji dio grafa slike 31 prikazuje grešku mini serije, izgladene verzije i grešku validacijskog skupa. Za treniranje mreže čiji su parametri definirani u poglavlju 6.4 u tablici 5, potrebno je bilo šest iteracija i te je samo treniranje trajalo oko 40 min na računalu. Vrsta računala koja se koristila pri izvođenju učenja je Lenovo, s procesorom Intel Core i3, RAM memorijom od 4 GB, tvrdim diskom od 300 GB i SSD diskom od 1 TB.



Slika 31. Proces treniranja neuronske mreže

Slika 32 prikazuje strukturu konvolucijske neuronske mreže koja se sastoji od tri konvolucijska sloja i potpuno povezanog sloja te će se u nastavku poglavlja svaki dio opisati kako bi se shvatila njihova uloga u treniranju neuronske mreže.



Slika 32. Blok shema pripreme za učenje optimizacijom parametara [21]

### 6.5.1. Konvolucijski blokovi

Konvolucijski skriveni slojevi duboke neuronske mreže imaju zadaću učenja značajki ulaznog signala definiranih u ulaznom sloju mreže. U ovom slučaju ulazni signal čine 7200 datoteka veličine 12800x1x3. Pomoću unaprijed određenog broja i veličine jezgri, faktora skaliranja i broja naslaganih blokova, konvolucijski slojevi pretvaraju ulazne signale u izlazne. Proces se sastoji od toga da se na ulazni signal primijeni taj određeni broj jezgri zatim se na svaku jezgru doda pomak te na kraju nelinearna aktivacijska funkcija kako bi se dobila izlazna matrica.

Nakon svakog konvolucijskog sloja, slijede slojevi normalizacije grupe (eng. *batch normalization*) i aktivacijske funkcije (ReLU) preko kojih se uvodi nelinearnost. Normalizacija grupe provodi se s ciljem smanjenja iznosa pomaka skrivenih slojeva jer se tijekom procesa učenja ulazne varijable signala množe (uz pomoć težina) i dodaju (uz pomoć pomaka) kako bi se uzrokovala aktivacije koje se potom vraćaju s gradijentima [21].

U ovom primjeru definirana su tri tipa konvolucijskih blokova (*KB1*, *KB2*, *KB3*).

### 6.5.2. Slojevi sažimanja

Slojevi sažimanja prema slici 33 nalaze se iza konvolucijskih blokova *KB1*, *KB2*, i *KB3* te njihova vrijednost iznosi [4 1] što je maksimalna moguća vrijednost. Razlog korištenja slojeva sažimanja u razvoju ovog modela, leži u njihovoj sposobnosti da modelu namaknu otpornost odnosno invarijantnost na translacije naučenih značajki unutar ulaznog signala [21]. Sažimanje smanjuje broj parametara koji se mogu učiti i kao posljedica smanjuje se mogućnost efekta pretreniranja.

### 6.5.3. Potpuno povezani sloj i SoftMax

Nakon što su značajke ulaznog vibracijskog signala otkrivene i izdvojene u konvolucijskim slojevima i zadnji sloj je sažiman maksimalnom vrijednošću, mape značajki se spajaju u jedan skup podataka. One se zatim postupkom poravnavanja transformiraju u jednodimenzionalni skup podataka. Takvi podaci čine ulaz u neurone koji se nalaze u potpuno povezanom sloju. Neuroni tijekom postupka treninga izvedu skalarni umnožak ulaznih podataka i zadane težine uz dodavanje pomaka, kako bi izračunao vrijednost izlaza, to jest omogućio ispravnu kategorizaciju klase u narednom SoftMax sloju [21].

## 6.6. Evaluacija, izračun i prikaz performansi novog modela

Evaluacija novih modela neuronskih mreža izračunava se pomoću ispravno klasificiranih uzoraka i ukupnog broja uzoraka prema sljedećoj jednadžbi (6.1):

$$\text{Uspješnost klasifikacije} = \frac{\text{Broj ispravno klasificiranih uzoraka}}{\text{Ukupan broj uzoraka}} \cdot 100 [\%]$$

6.1

Uspješnost klasifikacije izračunat će se u dvije faze:

- u fazi učenja modela gdje se neposredno nakon dobivanja modela određivanjem hiperparametra, klasificirani podaci uspoređuju s podacima za testiranje mreže prema poglavlju 6.3
- u fazi konačne evaluacije na testnom skupu podataka gdje se klasificirani podaci uspoređuju s testnim podacima.

Osim uspješnosti klasifikacije, koristiti će se i matrica zabune (eng. *confusion matrix*) ili matrica pogreške, koja u matričnom obliku prikazuje klasificirane podatke u odnosu na stvarne podatke.

| Matrica zabune   |                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Predviđena klasa | Stvarna klasa                                                                                                                                                                |
| 1000BBF          | 1000BBF 448 6.2% 0 0.0% 0 0.0% 0 0.0% 2 0.0% 0 0.0% 2 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 99.1% 0.9%                                                                     |
| 1000CBF          | 1000BBF 0 0.0% 450 6.3% 0 0.0% 0 0.0% 1 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 99.8% 0.2%                                                                     |
| 1000CRF          | 1000BBF 0 0.0% 0 0.0% 449 6.2% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%                                                                      |
| 1000ERF          | 1000BBF 0 0.0% 0 0.0% 1 0.0% 450 6.3% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 99.8% 0.2%                                                                     |
| 1000IMRF         | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%                                                                      |
| 1000IRBF         | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 443 6.2% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%                                                                      |
| 1000NS           | 1000BBF 1 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 99.8% 0.2%                                                              |
| 1000ORBF         | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%                                                               |
| 1500BBF          | 1000BBF 1 0.0% 0 0.0% 0 0.0% 0 0.0% 4 0.1% 0 0.0% 0 0.0% 447 6.2% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 98.9% 1.1%                                                                     |
| 1500CBF          | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%                                                                      |
| 1500CRF          | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 0 0.0% 0 0.0% 100% 0.0%                                                                      |
| 1500ERF          | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 0 0.0% 100% 0.0%                                                                      |
| 1500IMRF         | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 100% 0.0%                                                                      |
| 1500IRBF         | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 100% 0.0%                                                               |
| 1500NS           | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 1 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 99.8% 0.2%                                                              |
| 1500ORBF         | 1000BBF 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 450 6.3% 100% 0.0%                                                        |
|                  | 1000BBF 99.6% 0.4% 100% 0.0% 99.8% 0.2% 100% 0.0% 100% 0.0% 98.4% 1.6% 100% 0.0% 100% 0.0% 99.3% 0.7% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 99.8% 0.2% |

Slika 33. Matrica zabune

Matrica zabune na vizualan način predočava detaljan uvid u načine na koji klasifikacijski model griješi, to jest u koje skupine je klasificirao podatke testnoga skupa. U matrici zabune svaki redak predstavlja rezultate predviđanja za odgovarajuću klasu, dok svaki stupac predstavlja stvarnu klasu. Čelije na dijagonali definiraju postotak ispravno klasificiranih podataka s obzirom na pojedine klase, a čelije izvan dijagonale predstavljaju pogrešno klasificirana predviđanja. Zadnji sivi stupac predstavlja preciznost klasifikacijskog modela pri predviđanju svake klase, a zadnji sivi redak predstavlja postotak odaziva klasifikacijskog modela unutar svake klase. Drugim riječima, preciznost opisuje koliko često je predviđanje ispravno ukoliko se predvidi određena klasa, a odaziv mjeri koliko se ispravno predviđa određena stvarna klasa. Slika 34 prikazuje matricu zabune u fazi evaluacije na testnom skupu podataka modelu čija uspješnost klasifikacije nakon učenja iznosi 99,537 %. Uspješnost klasifikacije dobivena u fazi evaluacije, u kojemu se testni podaci uspoređuju s klasificiranim, iznosi 99,8 %. Dva , čija je stvarna klasa 1000 BBF, klasificirani su u 1000 NS i 1500 BBF. Jedan podatak, čija je stvarna klasa 1000 CRF, klasificiran je 1000 ERF. Najveći postotak krivo klasificiranih podataka pripada skupini 1000 IRBF u kojemu su 7 podataka krivo klasificirani. Dva podatka klasificirana su u skupinu 1000 BBF, jedan u 1000 CBF i četiri podatka u 1500 BBF. Zadnji podaci, koji su krivo klasificirani, su iz stvarne klase 1500 BBF. Dva podataka klasificirani su u klasu 1000 BBF i jedan podataka je klasificiran u klasu 1500 NS.

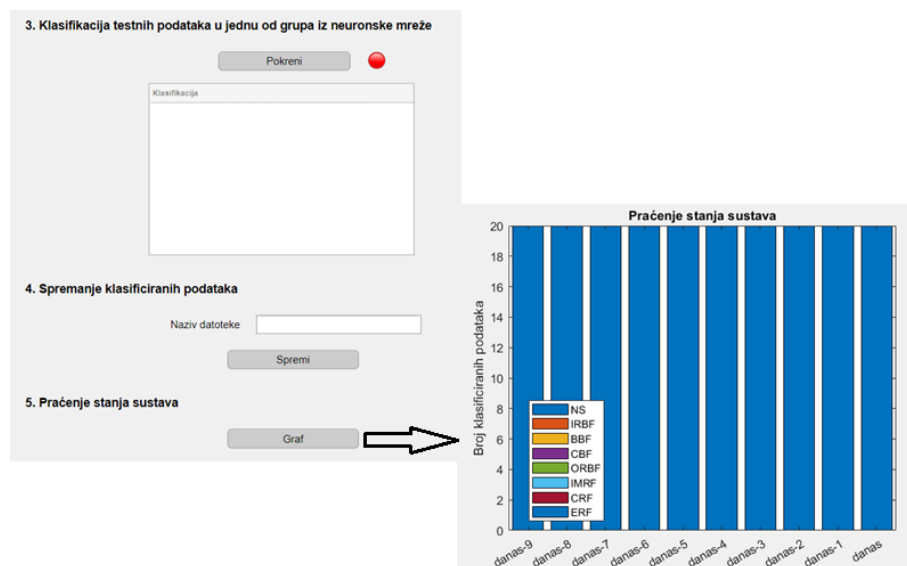
## 6.7. Učitavanje i primjena naučenog modela

Nakon učenja, evaluacije i pohrane modela, moguće je uspješno naučeni model primijeniti za klasifikaciju novih podataka koji se svakodnevno prikupljaju na senzorima montiranim na opremi.

Dakle, nakon učenja i evaluacije modela na poznatim klasama podataka, sljedeći korak je učitavanje jednog skupa podataka čije su klase nepoznate, ali ih je moguće izračunati. Samim time više nije moguće izračunati uspješnost klasifikacije s obzirom da nije poznata stvarna klasa učitanih podataka.

Kao i kod evaluacije modela, u primjeni modela je potrebno učitati željeni naučeni model zajedno s podacima koje je potrebno klasificirati. Slika 34 prikazuje tri sljedeće radnje koje se izvode u primjeni modela. U 3. koraku se nakon pritiska na tipku *Pokreni* odvija klasifikacija učitanih podataka nepoznate klase. Nakon toga je potrebno spremati takve podatke u datoteku i mjesto na računalu po vlastitom odabiru (4. korak). Takav postupak se može ponoviti kroz određeni period (npr. dnevni ili tjedni) kako bi se mogla vidjeti stvarna promjena stanja sustava.

Za takvo praćenje sustava moguće je dobiti graf prema slici 34, prema kojem je vidljivo kako se u periodu izračunavanja stanje opreme nije mijenjalo.



**Slika 34. Primjena modela i prikaz grafa**

Kako bi se jasnije prikazao način rada razvijenog grafičkog korisničkog sučelja, u poglavlju 6.9 će se simulirati primjer rada sučelja.

## 6.8. Kompajliranje

Kompajliranje je zadnji korak u praktičnom dijelu rada. MATLAB kompajler omogućuje korisnicima MATLAB razvojne okoline dijeljenje aplikacije s drugim zainteresiranim korisnima koji nemaju instaliranu MATLAB razvojnu okolinu. Prvi korak je u kompajleru izabrati korisničko sučelje koja predstavlja glavno sučelje pod nazivom „Avaluo“ što predstavlja izvedenicu od grčke riječi *ανάλυση* što znači analiza (Slika 37).



**Slika 35. Glavno sučelje**

Nakon odabira glavnog dijela korisničkog sučelje, moguće je popuniti dodatne informacije o aplikaciji poput imena autora i njegove osobne podatke, kratki sažetak i opis aplikacije (Slika 38).

Kako bi se aplikacija mogla pokrenuti, potrebno je priložiti sve datoteke koje su povezane s glavnim sučeljem. U ovom slučaju to su:

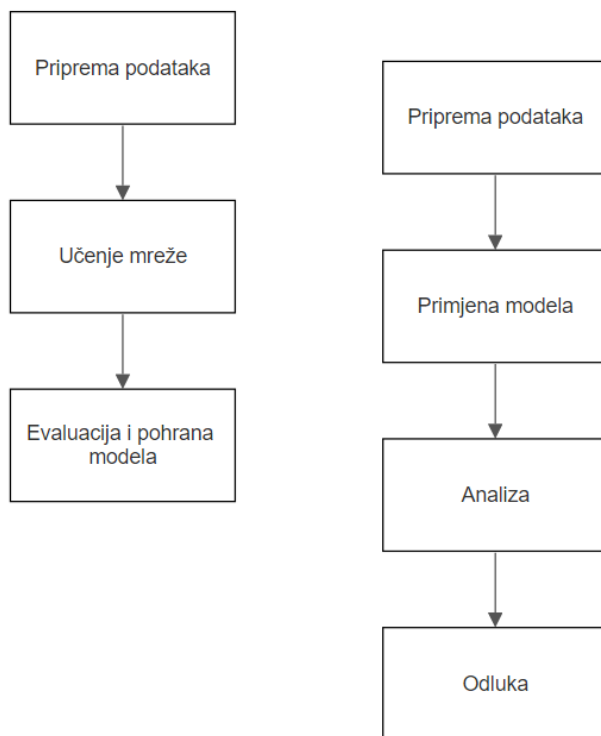
- slika koja se nalazi na glavnom sučelju (.png vrsta datoteke),
- dio sučelja za pripremu podataka (.mlapp vrsta datoteke),
- dio sučelja za učenje mreže (.mlapp vrsta datoteke),
- dio sučelja za evaluaciju modela (.mlapp vrsta datoteke),
- dio sučelja za primjenu modela (.mlapp vrsta datoteke)
- funkcija za čitanje datoteke podataka (.m vrsta datoteke),
- funkcija za zaustavljanje učenja neuronske mreže ukoliko ne dolazi do povećanja uspješnosti klasifikacije (.m vrsta datoteke).

Nakon uspješnog „pakiranja“ aplikacije, stvara se zasebna aplikacija na računalu u obliku .exe datoteke. Exe datoteka sadrži program koji je moguće izvršiti odnosno pokrenuti na

Windowsima. Ovim putem je moguće dijeliti aplikaciju s drugim Windows korisnicima, te za njezino pokretanje nije potrebno instalirati MATLAB razvojnu okolinu.

## 6.9. Primjer rada aplikacije

Razvijeno korisničko sučelje će primarno služiti dvjema procesima, prema dijagramima prikazanim na slici 36.



**Slika 36. Dijagrami procesa za početno i kontinuirano korištenje korisničkog sučelja**

Lijevi dijagram procesa opisuje prvotno korištenje korisničkog sučelja u kojemu se uz pomoć pripremljenih podataka, može naučiti mreža, evaluirati i pohraniti za daljnju upotrebu. Desni dijagram procesa prikazuje primjenu korisničkog sučelja za primjenu modela kod procjene stanja opreme uz pomoć novoprikupljenih podataka. U takvom procesu, učitavanjem naučenog i pohranjenog modela moguće je procijeniti stanje sustava i donijeti odluku.

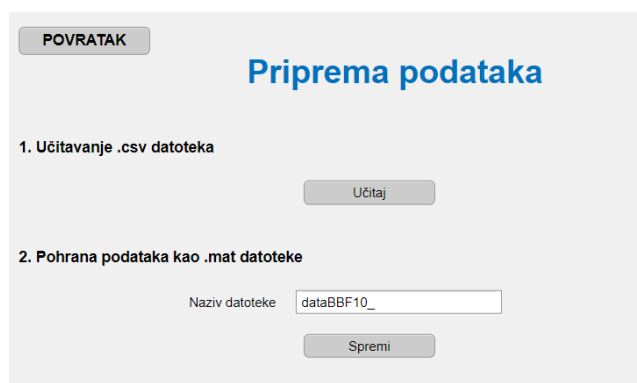
Prema slici 35, koja predstavlja glavno sučelje, potrebno je pritisnuti određenu tipku koja će voditi na jednu od sljedećih dijelova korisničkog sučelja kako bi se mogle izvršiti radnje u dijagramima procesa.

### 1. Priprema podataka za daljnji rad u sučelju

Prvi korak u pripremi podataka je učitavanje .csv datoteke (poglavlje 6.2.) s računala koja će se uz pomoć koda pretvoriti u .mat datoteku. Jedna datoteka predstavlja jedan tipa stanja sustava. S obzirom na veći broj datoteka koje će nastati pretvorbom .csv datoteke u drugom koraku je potrebno napisati željeni naziv datoteke (npr. dataBBF10\_



dataNS10\_, dataORBF15\_). Preporuča se ovakav naziv spremanja datoteka kako bi se moglo razlikovati koji je tip stanja sustava te koja je brzina okretaja. Nakon pritiska na tipku *Spremi*, potrebno je odrediti mjesto spremanja datoteka na računalu.



**Slika 37. Dio korisničkog sučelja za pripremu podataka**

Priprema podataka izvodi se svaki puta kad postoje novi podaci, odnosno prije samog učenja modela, ali i prije dnevnog postupka procjene stanja opreme.

## 2. Učenje mreže

Postupak učenja mreže provodi se najčešće pri uvođenju sustava u rad, ali može biti ponovljen u bilo kojem sljedećem trenutku, ukoliko postoje novi podaci za učenje.

Slika 39 prikazuje dio korisničkog sučelja koji se otvara nakon pritiska na tipku *Učenje mreže*. Kako bi bilo moguće naučiti neuronsku mrežu, svi prikupljeni podaci prethodno su razvrstani u mape s odgovarajućim imenima (Slika 30). Prva tablica prikazuje broj učitanih datoteka svake klase. U sljedećem koraku podjele podataka, moguće je odrediti koliki postotak učitanih podataka će pripadati podacima za treniranje mreže i za testiranje mreže te će ostatak podataka pripasti podacima za validaciju. U ovom primjeru podaci za treniranje mreže čine 60 %, podaci za testiranje mreže čine 30 % i podaci za validaciju čine 10 % od ukupnih podataka.

**POVRATAK**

## Učenje mreže

**1. Učitavanje potrebnih podataka**

**Učitaj podatke**

| Naziv grupe | Broj podataka |
|-------------|---------------|
| 1000_BBF    | 450           |
| 1000_CBF    | 450           |
| 1000_CRF    | 450           |
| 1000_ERF    | 450           |
| 1000_IMRF   | 450           |
| 1000_IRBF   | 450           |
| 1000_NS     | 450           |

**2. Podjela podataka**

**Podijeli**

Podaci za treniranje mreže

0.6

| Naziv grupe | Broj podataka |
|-------------|---------------|
| 1000_BBF    | 270           |
| 1000_CBF    | 270           |
| 1000_CRF    | 270           |
| 1000_ERF    | 270           |
| 1000_IMRF   | 270           |
| 1000_IRBF   | 270           |
| 1000_NS     | 270           |

Podaci za testiranje mreže

0.3

| Naziv grupe | Broj podataka |
|-------------|---------------|
| 1000_BBF    | 135           |
| 1000_CBF    | 135           |
| 1000_CRF    | 135           |
| 1000_ERF    | 135           |
| 1000_IMRF   | 135           |
| 1000_IRBF   | 135           |
| 1000_NS     | 135           |

Podaci za validaciju

0.1

| Naziv grupe | Broj podataka |
|-------------|---------------|
| 1000_BBF    | 45            |
| 1000_CBF    | 45            |
| 1000_CRF    | 45            |
| 1000_ERF    | 45            |
| 1000_IMRF   | 45            |
| 1000_IRBF   | 45            |
| 1000_NS     | 45            |

**Slika 38. Učitavanje i podjela podataka za učenje mreže**

Prikupljeni podaci za učenje mreže moraju nužno biti označeni odnosno klasa mora biti poznata.

Sljedeći korak (3.) je određivanje hiperparametra. U zagrada su navedeno rasponi za svaki hiperparametar, te nije moguće napisati broj koji nije u tom rasponu. U 4. koraku nakon pritiska na tipku *Pokreni*, započinje učenje mreže te se pojavljuje graf prema slici 31., prema kojemu je moguće pratiti na koji način se odvija učenja mreže. Nakon što je učenje mreže gotovo, lampa koja je inače crvene boje će poprimiti zelenu boju. U 5. koraku se odvija klasifikacija podataka za testiranje (poglavlje 6.2.). U tablici je prikazana stvarna klasa u odnosu na predviđenu klasu, a u polju *Točnost mreže* isčitava se uspješnost klasifikacije prema formuli 6.1. U 6. koraku, nakon pritiska na tipku *Spremi*, potrebno je odrediti mjesto spremanja datoteke (Slika 40). Datoteka koja će nastati će imati .mat format. Sadržavat će naučenu mrežu, opcije koje su se koristile prilikom učenja mreže i vrijednost pogreške naučene mreže.

### 3. Određivanje hiperparametra

Broj naslaganih blokova (0 - 8)

Broj jezgri (4 - 24)

Momentum (0.80 - 0.95)

Veličina jezgri (4 - 64)

Koeficijent učenja (0.001 - 0.01)

Faktor skaliranja(1 - 3)

### 4. Pokretanje treniranja mreže

Pokreni

### 5. Evaluacija mreže

Pokreni

|   | Stvarna klasa |
|---|---------------|
| 1 | 1000_BBF      |
| 2 | 1000_BBF      |
| 3 | 1000_BBF      |
| 4 | 1000_BBF      |
| 5 | 1000_BBF      |
| 6 | 1000_BBF      |
| 7 | 1000_BBF      |

Točnost mreže

### 6. Spremanje mreže

Spremi

**Slika 39. Dio sučelja za određivanja hiperparametara, učenje, evaluaciju i spremanje mreže**

### 3. Evaluacija modela

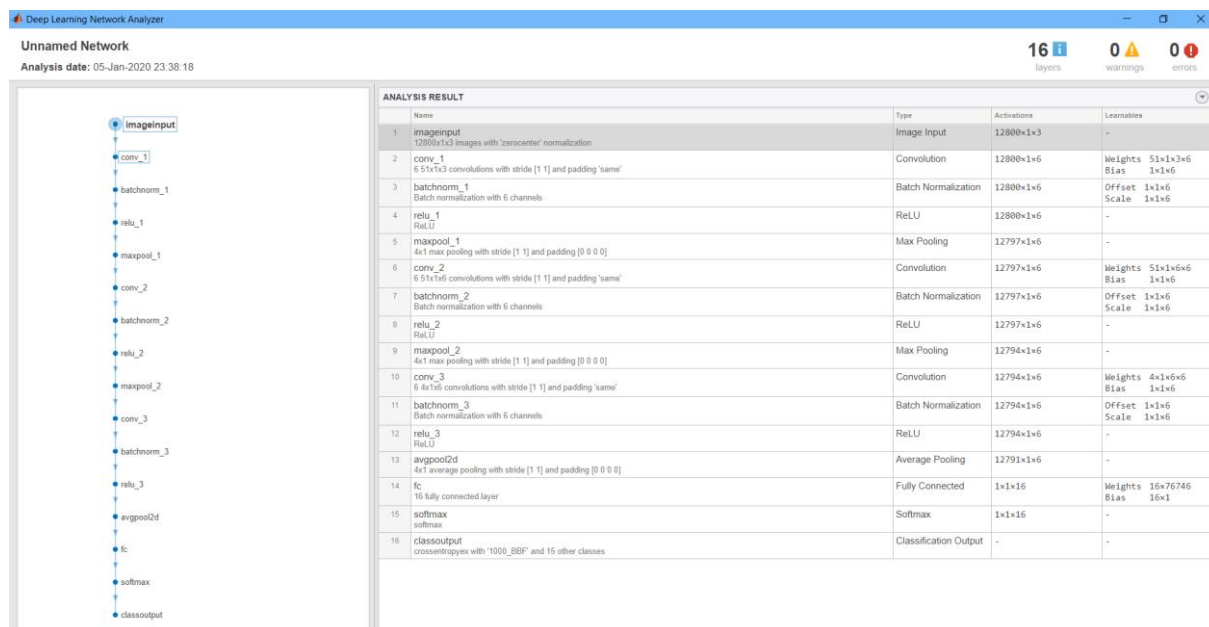
```
graph TD; A[POVRATAK] --> B[Evaluacija modela]; B --> C[1. Učitavanje naučenog modela]; C --> D[Učitavanje podataka]; D --> E[3. Klasifikacija testnih podataka u jednu od grupa iz neuronske mreže]; E --> F[4. Prikaz matrice zabune]; F --> G[5. Analiza neuronske mreže]; G --> A;
```

The diagram illustrates the evaluation process flow. It begins with a 'POVRATAK' (Return) button, which leads to the 'Evaluacija modela' (Model Evaluation) section. This section contains two main steps: '1. Učitavanje naučenog modela' (Loading the trained model) and '2. Učitavanje testnih podataka za ispitivanje mreže' (Loading test data for network testing). Step 1 includes a 'Učitavanje modela' (Load model) button. Step 2 includes a 'Učitavanje podataka' (Load data) button. Following these steps, the process moves to '3. Klasifikacija testnih podataka u jednu od grupa iz neuronske mreže' (Classification of test data into one of the groups from the neural network). This step includes a 'Pokreni' (Run) button and a table with two columns: 'Stvarna vrijednost' (Actual value) and 'Klasifikacija' (Classification). Below the table is a text input field with the value '0'. The process then continues to '4. Prikaz matrice zabune' (Display confusion matrix), which includes a 'Prikaži' (Show) button. Finally, it reaches '5. Analiza neuronske mreže' (Neural network analysis), which includes an 'Analiza mreže' (Analyze network) button. The flow concludes by returning to the 'POVRATAK' button.

### Slika 40. Evaluacija modela

Slika 41 prikazuje sve korake koje i radnje koje je moguće izvesti u dijelu korisničkog sučelja za evaluaciju modela. U 1. i 2. koraku je potrebno učitati željeni prethodno naučeni model s podacima koji se žele klasificirati te čija je stvarna klasa poznata. S obzirom da se učitavaju poznate klase podataka i u ovom dijelu sučelja moguće je dobiti, kolika je uspješnost klasifikacije. Lampe služe kako bi znali u kojem trenutku je učitavanje podataka za ispitivanje mreže gotovo te njihova klasifikacija. U 4. koraku je

moгуće na vizualan nać predoditi uspješnost klasifikacije uz pomoć matrice zabune (poglavlje 6.6.). Novije verzije MATLAB-a omogućuju analizu naućenih modela. U analizi je omoguće saznati dimenziju ulaznih podataka, od koliko slojeva se sastoji mreža te što sve ćini pojedini sloj .



Slika 41. Analiza mreže

#### 4. Primjena naućenih modela

Primjena naućenih modela na novim podacima predstavlja uobićajenu rutinu pri određivanju stanja opreme na temelju novo prikupljenih podataka sa senzora.

Simuliran je primjer u kojem se prikupljaju podaci sa senzora i isti klasificiraju naućenim modelom pomoću razvijenog grafićkog korisnićkog sućelja u periodu od 10 dana rada sustava. Svaki dan prikuplja se 20 mjerenja, a u svakom mjerenju prikuplja se 12800 podataka sa troosnog senzora akceleracije montiranog na opremi. Prikupljeni podaci svakodnevno se ućitavaju u korisnićko sućelje zajedno s naućenim modelom koji ima najveći postotak uspješnosti klasifikacije, a zatim klasificiraju u određene klase uz pomoć ućitanog modela (Slika 42).

**2. Učitavanje podataka**

Učitavanje podataka

| Naziv grupe | Broj podataka |
|-------------|---------------|
| Podaci 2    | 20            |

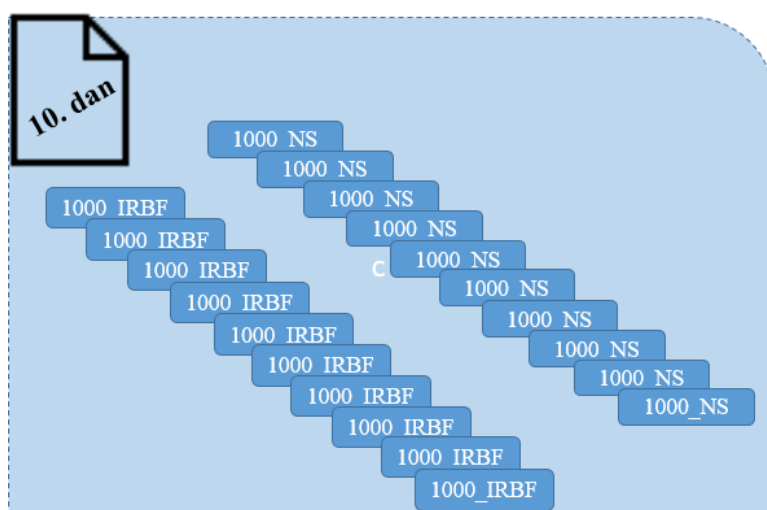
**3. Klasifikacija testnih podataka u jednu od grupa iz neuronske mreže**

Pokreni

|   | Klasifikacija |
|---|---------------|
| 1 | 1000_IRBF     |
| 2 | 1000_IRBF     |
| 3 | 1000_IRBF     |
| 4 | 1000_NS       |
| 5 | 1000_NS       |
| 6 | 1000_NS       |
| 7 | 1000_NS       |
| 8 | 1000_NS       |

**Slika 42. Učitavanja novih podataka i klasifikacija**

Nakon svake klasifikacije podaci se spremaju u obliku .mat datoteke koja će se kasnije koristiti za stvaranje grafa, a kako bi se na vizualan način moglo pratiti stanje sustava. Slika 43 prikazuje primjer jedne takve spremljene datoteke za određeni dan.



**Slika 43. Primjer sadržaja spremljene datoteke**

Na primjeru vidljivom na slici, 20 podataka klasificirano je u dvije skupine. Deset podataka klasificirano je u skupinu 1000 NS, a drugih deset podataka klasificirano je u 1000 IRBF. Nakon provedene simulacije za deset dana, sveukupno postoji deset takvih datoteka s klasificiranim podacima. Tablica 6 prikazuje sadržaj svake od tih pojedinih datoteka, pri čemu vrijednosti u tablici prikazuju broj klasificiranih podataka pojedinog stanja.

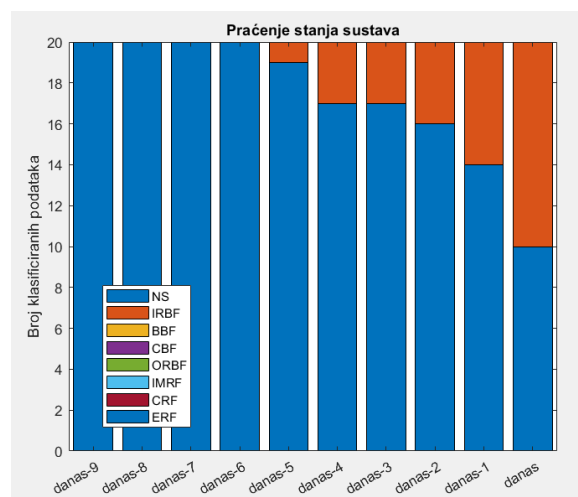
Tablica 6. Sadržaj spremljenih datoteka koje služe za praćenje sustava

| Tip stanja/Dani | 1. dan | 2. dan | 3. dan | 4. dan | 5. dan | 6. dan | 7. dan | 8. dan | 9. dan | 10. dan |
|-----------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| 1000 NS         | 20     | 20     | 20     | 20     | 19     | 17     | 17     | 16     | 14     | 10      |
| 1000 IRBF       | 0      | 0      | 0      | 0      | 1      | 3      | 3      | 4      | 6      | 10      |

Pojedinačnim praćenjem svakodnevnih podataka stručnjak ne može uvidjeti dolazi li do promjene u načinu rada sustava. Ukoliko postoji velika količina prikupljenih podataka, manji broj klasificiranih podataka u klase koje označavaju pojedine kvarove sustava, ne mora nužno značiti da se sustav počeo kvariti. Moguće je da je model krivo klasificirao podatke bez obzira na visoku uspješnost klasifikacije. Zato treba oprezno pristupiti tome problemu, te na kontinuiran način pratiti stanje sustava tj. pratiti trendove u vremenskim okvirima.

U razvijenom grafičkom korisničkom sučelju, omogućeno je učitavanje većeg broja (unazad 10 dana) prethodno pohranjenih rezultata klasifikacije prikupljenih podataka i prikaz istih u grafičkom obliku.

Za dobivanje grafa na slici 44, učitane su datoteke koje sadržavaju skupine klasificiranih podataka iz skupa za testiranje na određene dane. Nakon učitavanja datoteka, zadana je funkcija koja prebrojava ukupan broj pojedinačnih skupina po datotekama.



Slika 44. Grafički prikaz procenjenjenih klasa za pojedine dane

Prema grafu, u prva četiri dana svi podaci klasificirani su u klasu *normalnog stanja* čime je moguće zaključiti kako sustav i dalje pravilno radi, ne javljaju se naznake kvarova i nije potrebno donositi nikakve odluke prilikom održavanja sustava. Peti dan jedan od podataka klasificiran je u klasu IRBF. To može ukazivati na početak razvoja kvara, ali

s obzirom da je broj klasificiranih podataka u klasu IRBF u manjini, stručnjak koji analizira podatke može pričekat sljedeći dana i novu klasifikaciju podataka. U sljedećih nekoliko dana broj podataka koji su klasificirani u klasu IRBF povećava se. To ukazuje kako se stanje sustava pogoršava te se javlja kvar koji će utjecati na sustav. U ovakvom slučaju stručnjaci bi morali prepoznati o kojoj vrsti kvara se radi i napraviti plan održavanja. U stvarnosti to neće biti tako jednostavno. Naime, svaki sustav je specifičan te stvaranje idealnog modela koji će imati veliku uspješnost klasifikacije podataka zahtjeva novi skup podataka označenih za potrebe učenja odnosno novo učenje modela. Problemi koji se mogu javiti u stvarnosti, njihova pojašnjenja i moguća rješenja prikazani su u tablici 6.

**Tablica 7. Mogući problemi i rješenja pri upotrebi korisničkog sučelja**

| Problemi                                   | Pojašnjenje                                                                                                                                                                       | Moguća rješenja                                                                                                                                                                                                                                                 |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nepoznavanje vrste kvarova</b>          | Kvarovi utječu na sustave na različite načine. Pojedinačni kvar može utjecati na zatajenje samo jednog dijela sustava, dok drugi kvar može utjecati na potpuno zatajenje sustava. | Stručnjaci koji se bave analizama moraju osim analitičkih sposobnosti, znati na koji način pojedini kvarovi utječu na sustav. Tako da je potrebno još u ranim fazama rada sustava analizirati utjecaj kvarova na sustav (FTA, FMEA analiza).                    |
| <b>Vrijeme provođenja plana održavanja</b> | Na grafu prema slici 44 moguće je vidjeti povećanje klasificiranih podataka kroz nekoliko dana. Potrebno je odrediti pravo vrijeme provođenja radnji održavanja.                  | Stručnjaci bi iskustveno trebali znati u kojem trenutku je potrebno provesti plan održavanja. Također je moguće odrediti prag postotka klasificiranih podataka u korisničkom sučelju, koji kada se prijeđe sučelje samo javlja da je potrebno izvršiti određene |

|  |  |                                                       |
|--|--|-------------------------------------------------------|
|  |  | radnje kako bi se sustav<br>vratio u normalno stanje. |
|--|--|-------------------------------------------------------|



## 7. ZAKLJUČAK

Integrirani sustav praćenja stanja i dijagnostičke strategije postale su važan dio životnog ciklusa raznih sustava. Sposobnost otkrivanja anomalija, analiziranje kvarova i predviđanje budućih stanja, samo su neki od zadataka održavanja po stanju koji su mogući uz pomoć novih razvijenih tehnologija. Danas, većina sustava generira velike količine podataka koje je moguće prikupiti uz pomoć senzora. Ti podaci mogu služiti za učenje raznih modela, koji su navedeni u poglavlju za praćenje i dijagnostiku kvarova, i pomoći pri donošenju odluka o održavanju sustava. Statističke metode, koje služe kao podloga u dijagnostici temeljenoj na podacima, sve se više nadopunjavaju s modernim tehnikama temeljenim na algoritmima strojnog i dubokog strojnog učenja, koje imaju sposobnost učenja iz dostupne velike količine podataka. Stručnjaci razvijaju metode pomoću kojih je moguće naučiti modele koji će moći s velikom uspješnosti klasificirati podatke u određene skupine. Za izradu praktičnog rada, potrebno je bilo u zasebnom poglavlju opisati jednu od metoda dubokog strojnog učenja koja se naziva konvolucijska neuronska mreža. Ona koristi postupak konvolucije, klasificirajući ulazni signal najčešće u obliku slika ili senzora, u neku od predefiniranih klasa.

Uz dobivene podatke, prikupljene simuliranjem različitih kvarova na eksperimentalnom simulacijskom postavu, koristila su se prethodna teorijska znanja o strojnom učenju pod nadzorom i konvolucijskoj neuronskoj mreži pri razvoju jedinstvenog korisničkog sučelja. Razlog razvoja korisničkog sučelja je taj što trenutno ne postoji jedinstveno korisničko sučelje koje objedinjuje module za prikupljanje i tumačenje podataka te integraciju algoritma za strojno učenje.

Uz pomoć korisničkog sučelja razvijenog u okolini MATLAB, moguće pripremiti podatke za njihovo korištenje u programu, odabrati hiperparametre i naučiti neuronsku mrežu, evaluirati model te ga primijeniti na novom skupu podataka. Takvo korisničko sučelje, koje se naziva „Avaluan“, moguće je podijeliti s drugim korisnicima koji se ne služe MATLAB razvojnom okolinom pomoću opcije kompajliranja. Nakon kompajliranja nastala je .exe datoteka koju je moguće koristiti u Windows sustavu.

U konačnici, korisničko sučelje omogućuje primjenu naučenih modela na novim skupovima podataka što može utjecati na odluke stručnjaka prilikom stvaranja plana i programa za održavanje. Na temelju objašnjenog primjera primjene u stvarnoj okolini, može se zaključiti na koji način primjenom razvijenog korisničkog sučelja stručnjaci održavanja mogu pravovremeno shvatiti što se događa u sustavu na temelju klasificiranih podataka te kako

određeni kvarovi utječu na rad sustava. Ovakvo korisničko sučelje moguće je dalje nadograđivati, kako bi se dobilo još više informacija. U daljnjem razvoju, bilo bi potrebno razmotriti mogućnost implementacije tehnika strategije preskriptivnog održavanja, prema kojem bi računalni sustav na temelju podataka o stanju sustava imao mogućnost predlagati radnje koje je potrebno napraviti, kako bi se oprema zadržala u ispravnom radnom stanju.

---

**LITERATURA**

- [1] R. Keith Mobley: An Introduction to Predictive Maintenance, 2. edition. Elsevier Science, 2002.
- [2] D. Lisjak, D. Kolar, M. Tošić: Održavanje, ver. 8.1, Fakultet strojarstva i brodogradnje, Zagreb
- [3] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, B. Wu: Intelligent Fault Diagnosis and Prognosis for Engineering Systems. John Wiley & Sons, Inc., Hoboken, New Jersey 2006.
- [4] D. Romahadi, A. A. Luthfie, L. B. Desti Dorion: Detecting classifier-coal mill damage using a signal vibration analysis. Universitas Mercu Buana, 2019.
- [5] A. Mosallam, K. Medjaher, N. Zerhouni: Component based Data-driven Prognostics for Complex Systems: Methodology and Application. PHM Conference Beijing, listopad 2015.
- [6] B. Huang, Y. Di, C. Jin, J. Lee: Review od data-driven prognostics and health management techniques: Lessons learned from PHM data Challenge Competitions. Machine Failure Prevention Technology, 2017.
- [7] E. Ruijters, M. Stoelinga: Fault tree analysis: A survey of the state of the art in modeling, analysis and tools. Computer science review, svibanj 2015.
- [8] J. Lee, C. Jin, Z. Liu, H.D. Ardakani: Introduction to Data-Driven Methodologies for Prognostics and Health Management. Springer International Publishing AG, 2017.
- [9] G.W. Vogl, B.A. Weiss, M. Helu: A Review of diagnostic and prognostic capabilities and best practices for manufacturing. Springer Science and Business Media New York, 2016.
- [10] Z. Gao, C. Cecati, S.X. Ding: A Survey of Fault Diagnosis and Fault Tolerant Techniques – Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches. IEEE Transactions on Industrial Electronics, 2015.
- [11] T. Sutharssan, S. Stoyanov, C. Bailey, C. Yin: Prognostic and health management for engineering systems: A review of the data-driven approach and algorithms. The Journal of Engineering, 2015.
- [12] J. Harmouche, C. Delpha, D. Diallo: Incipient fault detection and diagnosis based on Kullback-Leibler divergence using principal component analysis: Part II. Signal Processing, 2014.

- [13] Y. Song, Q. Jiang, X. Yan: Fault diagnosis and process monitoring using a statistical pattern framework based on a self-organizing map. *Journal of Central South University*, 2015.
- [14] S. Arora, W. Hu, P.K. Kothari: An Analysis of the t-SNE Algorithm for Data Visualization. *COLT*, 2018.
- [15] K. Tidriri, N. Chatti, S. Verron, T. Tipica: Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. Elsevier Science, ožujak 2016.
- [16] Z. Yin, J. Hou: Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes. *Neurocomputing*, siječanj 2016.
- [17] F. Kang, S. Han, R. Salgado, J. Li: System probabilistic stability analysis of soil using Gaussian process regression with Latin hypercube sampling. *Computers and Geotechnics*, Elsevier, siječanj 2015.
- [18] B. Cai, et al.: A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks, *Mech. Syst. Signal Process. Mechanical Systems and Signal Processing*, Elsevier. 2016.
- [19] H. Zhou, J. Chen, G. Dong, R. Wang: Detection and diagnosis of bearing faults using shift-invariant dictionary learning and hidden Markov model. *Mechanical Systems and Signal Processing*, Elsevier. 2016.
- [20] Y. LeCun, Y. Bengio, G. Hinton: Deep Learning. *Nature Publishing Group*. svibanj 2015.
- [21] D. Kolar: Model rane procjene kvarova rotacijske opreme primjenom dubokog strojnog učenja, 2019.
- [22] P. Kim: MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence. Apress. lipanj 2017.
- [23] Ž. Ban, J. Matuško, I. Petrović: Primjena programskog sustava MATLAB za rješavanje tehničkih problema, Zagreb, Graphis d.o.o., 2010.
- [24] Internet stranica: <https://www.mathworks.com/products/matlab/app-designer.html> (pristupljeno 27. studeni 2019.)

---

## **PRILOZI**

### **I. CD-R**