

# Primjena strojnog učenja u održavanju

---

**Doko, Marin**

**Master's thesis / Diplomski rad**

**2018**

*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://um.nsk.hr/um:nbn:hr:235:655932>

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

*Download date / Datum preuzimanja:* **2024-07-24**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Marin Doko**

Zagreb, 2018.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

Mentori:

Prof. dr. sc. Dragutin Lisjak, dipl. ing.

Student:

Marin Doko

Zagreb, 2018.

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

Zahvaljujem se:

Mentoru prof. dr. sc. Dragutinu Lisjaku na ukazanom povjerenju, pomoći i savjetima tijekom izrade rada,

Kolegi asistentu Davoru Kolaru na pomoći i literaturi pruženoj tijekom izrade rada,

Svojim roditeljima na tome što su mi omogućili završetak studija,

Marijani na ljubavi i potpori!

Marin Doko



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 Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	
Ur. broj:	

## DIPLOMSKI ZADATAK

Student: **MARIN DOKO** Mat. br.: 0035186650

Naslov rada na hrvatskom jeziku: **Primjena metoda strojnog učenja u održavanju**

Naslov rada na engleskom jeziku: **Application of machine learning methods in maintenance**

Opis zadatka:

Današnji stupanj razvoja informacijskih tehnologija u domeni održavanja tehničkih sustava omogućava prikupljanje velike količine podataka vezanih za objekte održavanja koji su nadzirani raznim senzorskim sustavima. Zbog velike količine podataka te nedostatka vremena analiza i tumačenje istih realno nije više moguća ručnom obradom te se sve više primjenjuju metode strojnog učenja.

U skladu s navedenim, u radu je potrebno obraditi sljedeće:

1. opisati strategiju Održavanje 4.0.
2. opisati tehnologiju "Internet stvari"
3. prikazati mogućnosti dostupnih platformi strojnog učenja
4. opisati mogućnosti najčešće korištenih algoritama strojnog učenja
5. primjenom senzorskog sustava prikupiti veliku količinu podataka s nekog objekta održavanja te iste pripremiti za strojnu obradu
6. na temelju eksperimenta, predložiti najučinkovitiji algoritam strojnog učenja kojim se opisuje trenutno "zdravstveno stanje" u kojem se objekt održavanja nalazi
7. dati zaključak.

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

Zadatak zadan:  
03. svibnja 2018.

Rok predaje rada:  
05. srpnja 2018.

Predvideni datum obrane:  
11. srpnja 2018.  
12. srpnja 2018.  
13. srpnja 2018.

Zadatak zadao:

prof. dr. sc. Dragutin Lisjak

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

## SADRŽAJ

1. UVOD.....	1
2. ODRŽAVANJE STROJNE OPREME .....	2
2.1. Strategije održavanja.....	2
2.1.1. Korektivno održavanje.....	3
2.1.2. Preventivno - plansko održavanje.....	3
2.1.3. Prediktivno održavanje .....	5
2.2. Održavanje po stanju.....	6
2.3. Mjerenje i analiza vibracija.....	7
2.4. „Internet stvari“ – primjena u industriji .....	11
2.5. Strategija <i>Industrija &amp; Održavanje 4.0</i> .....	13
3. STROJNO UČENJE.....	17
3.1. Neuronske mreže.....	21
3.1.1. Model neuronske mreže .....	22
3.1.2. Učenje neuronske mreže .....	25
3.1.3. Algoritam povratnog prostiranja pogreške .....	28
3.1.4. Regularizacija neuronske mreže .....	29
3.2. Povratne neuronske mreže .....	30
3.3. Autoenkoder.....	32
3.4. Ograničeni Boltzmannovi strojevi .....	33
3.5. Duboke konvolucijske neuronske mreže .....	34
4. EKSPERIMENTALNI POSTAV ZA SIMULACIJU KVAROVA ROTACIJSKE OPREME .....	40
5. INTELIGENTNI SUSTAV PROCJENE ZDRAVSTVENOG STANJA STROJNE OPREME .....	46
5.1. Učitavanje snimljenih podataka unutar programskog okruženja MATLAB .....	49
5.2. Odabir hiperparametara i treniranje konvolucijske neuronske mreže.....	49
5.3. Rezultati testiranja konvolucijske neuronske mreže .....	53
6. ZAKLJUČAK.....	57
LITERATURA.....	58

**POPIS SLIKA**

Slika 1	Taksonomija strategija održavanja [2] .....	3
Slika 2	Krivulja „kade“ – dijagram učestalosti i vrsti kvarova .....	4
Slika 3	Tipični koraci pri održavanju po stanju [6] .....	7
Slika 4	Unutrašnjost tipičnog piezoelektričnog akcelerometra [5] .....	9
Slika 5	Odnos između vremenske domene i frekvencijske domene [1] .....	10
Slika 6	Dijagram općenitog pristupa integriranoj implementaciji Prediktivnog Održavanja 4.0. Adaptirano iz [9] .....	16
Slika 7	Razlika između klasifikacije i regresije .....	18
Slika 8	Primjer rezultata algoritma detekcije anomalije. N1 i N2 su grupirana normalna stanja, dok O stanja označavaju odstupanja od normalnih stanja. [15] .....	19
Slika 9	Slikoviti prikaz biološkog neurona (lijevo) i poveznica s umjetnim neuronom (desno). [13] .....	21
Slika 10	Razlike između sigmoidalne funkcije (lijevo) i step funkcija korištene u prvom modelu perceptrona (desno) .....	23
Slika 11	Struktura neuronske mreže koja odgovara Jednadžbi (8) [16] .....	24
Slika 12	Prikaz zglobne aktivacijske funkcije RELu (lijevo) i tangens hiperbolne funkcije (desno) .....	25
Slika 13	Grafički prikaz odnose funkcije cilja i vektora težina s dvije vrijednosti. [16] ....	26
Slika 14	Različiti odabiri stope učenja utječu na konvergenciju rješenja .....	27
Slika 15	Primjer učenja dvoslojne neuronske mreže trenirane na 10 trening podataka iz sinus funkcije. M označava broj neurona u skrivenom sloju. Podnaučenost lijevo, generalizacija u sredini, prenaučenost desno. ....	30
Slika 16	Utjecaj faktora regularizacije $\lambda$ na kapacitet modela .....	30
Slika 17	Struktura povratne neuronske mreže [22] .....	31
Slika 18	Struktura autoenkodera [24] .....	32
Slika 19	Dijagram rekonstrukcija signala unutar autoenkodera .....	33
Slika 20	Arhitektura a) Ograničenog Boltzmannovog stroja, b) Duboke probabilističke mreže, c) Dubokog Boltzmannovog stroja [22] .....	34
Slika 21	Povezivanje lokalnih receptivnih polja i skrivenih neurona [33] .....	36
Slika 22	Primjer strukture konvolucijske neuronske mreže LeNet-5 iz [28] .....	38
Slika 23	Simulator kvarova. 1 - Trofazni elektromotor, 2 – Frekventni regulator, 3 – spojka, 4 – modul za simuliranje debalansa (osovina, ležajevi, teret), 5 – 3-osni akcelerometar .....	40
Slika 24	Dijagram tijeka provođenja eksperimenta prikupljanja vibracija .....	43
Slika 25	Izgled sučelja za snimanje podataka vremenske i frekvencijske domene na računalo .....	44
Slika 26	Struktura sustava za upravljanje zdravstvenim stanjem strojne opreme. Adaptirano prema [34]. ....	46
Slika 27	Kategorizacija metoda procjene zdravstvenog stanja sustava temeljenih na umjetnoj inteligenciji. Adaptirano iz [24]. ....	47
Slika 28	Razlika između a) konvencionalnih metoda strojnog učenja b) dubokog učenja [24] .....	48
Slika 29	Učitavanje tablične datoteke unutar MATLAB-a .....	49
Slika 30	Slojevi mreže unutar MATLAB-a .....	51
Slika 31	Točnost klasifikacije tijekom učenja. Epohe su prikazane bijelim i sivim stupcima. ....	52
Slika 32	Funkcija gubitka kroz epohe učenje .....	52

---

Slika 33	Vizualizacija automatskog učenja značajki kroz slojeve konvolucijske neuronske mreže pomoću t-SNE algoritma .....	54
Slika 34	Primjer izmjerenih vibracija jednog mjerenja ekscentričnog rotora .....	55
Slika 35	Prikaz kernela prvog konvolucijskog sloja u vremenskoj domeni .....	56
Slika 36	Prikaz kernela prvog konvolucijskog sloja u frekvencijskoj domeni .....	56



---

**POPIS TABLICA**

Tablica 1	Četiri sloja arhitekture tehnologije Internet stvari [8] .....	12
Tablica 2	Dimenzije konvolucijskog sloja i pripadajućih hiper-parametara.....	38
Tablica 3	Dimenzije sažimajućeg sloja i pripadajućih hiper-parametara.....	39
Tablica 4	Moduli za simuliranje kvarova na rotacijskoj opremi.....	41
Tablica 5	Mjerni senzori.....	42
Tablica 6	Karakteristike sustava za akviziciju podataka.....	42
Tablica 7	Parametri eksperimenta .....	43
Tablica 8	Odabrana struktura i parametri konvolucijske neuronske mreže .....	51
Tablica 9	Parametri učenja .....	52

---

**SAŽETAK**

Inteligentni sustavi rane dijagnostike kvarova unutar sustava održavanja strojne opreme sve više zamjenjuju konvencionalne metode dijagnostike koje uvelike ovise o ljudskom faktoru. Primjenom algoritama strojnog učenja unutar strategije prediktivnog održavanja povećava se efikasnost sustava, kvaliteta proizvoda i usluge te se smanjuje vrijeme zastoja rada. U ovom radu primijenjen je algoritam konvolucijskih neuronskih mreža za analizu vibracijskih signala prikupljenih sa senzora na eksperimentalnom postavu rotacijske opreme. Vibracijski signal ulazi u neuronsku mrežu bez prethodnih operacija ručnih izvlačenja značajki te kroz višeslojnu strukturu mreže prolazi kroz niz operacija kojima se značajke signala automatski uče, izlučuju i grupiraju. Točnost klasifikacije ovakvog algoritma iznosi oko 99.6% i time pokazuje potencijal unutar dijagnostike kvarova strojne opreme i nameće se kao centar daljnjeg istraživanja.

**Ključne riječi:** Prediktivno održavanje, strojno učenje, Internet stvari, konvolucijske neuronske mreže, vibracije

---

**SUMMARY**

Intelligent fault diagnosis systems are replacing conventional diagnostic methods which are dependent on commonly unreliable human factor. System efficiency, products and services quality can all be increased by applying machine learning algorithms in predictive maintenance strategy and, in the same time, machine downtime can be minimized. In this thesis, Convolutional neural networks are applied for analysis of vibration signals, which are measured from experimental setup of rotational equipment. Vibration signal is imported in CNN without manual extraction of features and passed through multi-layer architecture where the features are automatically learned, extracted and grouped. Classification accuracy of this proposed algorithm is 99.6%, so it shows large potential for appliance in fault diagnostics and prediction field and it is becoming central part of further investigations.

Key words: Predictive maintainance, machine learning, Internet of things, convolutional neural networks, vibrations

## 1. UVOD

U današnje vrijeme velikog i ubrzanog tehnološkog razvoja, umjetna inteligencija sve više nailazi na primjenu u raznim granama ljudskih aktivnosti. Umjetna inteligencija se većinom povezuje s humanoidnim robotima i sličnim inovacijama te je primjena umjetne inteligencije unutar tehnoloških poduzeća još uvijek nedovoljno istraženo područje djelovanja. Održavanje imovine može činiti ključnu razliku u uspješnosti određenog poduzeća, poglavito u manufakturno – industrijskim poduzećima te prijevozničkim sustavima, kao što su avionski prijevoznici ili sustavi javnog prijevoza u gradovima ili državama. Neadekvatne tehnike održavanja imovine poduzeća mogu značajno utjecati na ukupne troškove poduzeća, zadovoljstvo radnika te konačnu uslugu poduzeća i zadovoljstvo korisnika usluga. Slijedom toga, nameće se izazov primjene najnovijih strategija održavanja i tehnika obrade podataka u cilju povećanja vijeka trajanja imovine, smanjenja vremena zastoja opreme te poboljšavanja kvalitete proizvoda ili usluge.

Ističe se upotreba strategije prediktivnog održavanja unutar koje su integrirane razne metode umjetne inteligencije, uključujući hardverska i softverska sučelja, kako bi se na vrijeme predvidjeli kvarovi na opremi i spriječili nepoželjni zastoji rada. Tako prediktivno održavanje uključuje upotrebu tehnologije „Internet stvari“ koja povezuje fizička postrojenja s virtualnim aplikacijama te softverska rješenja u vidu primjene algoritama strojnog učenja na prikupljene podatke sa senzora na uređajima. Rezultati prediktivnog održavanja mogu biti detekcije vrlo ranih nastajanja kvarova, dijagnoza i lokalizacija kvara te predikcija korisnog vijeka trajanja.

Unutar ovog rada u drugom poglavlju opisane su dosadašnje strategije održavanja strojne opreme, opisana je tehnika mjerenja i analize vibracija kao metodu unutar održavanja po stanju te su opisani pojmovi Prediktivnog Održavanja 4.0 kao i smjernice za integraciju unutar postrojenja i tehnologija „Internet stvari“.

U trećem poglavlju opisane su neke od najkorištenijih metoda strojnog učenja kao što su neuronske mreže i konvolucijske neuronske mreže.

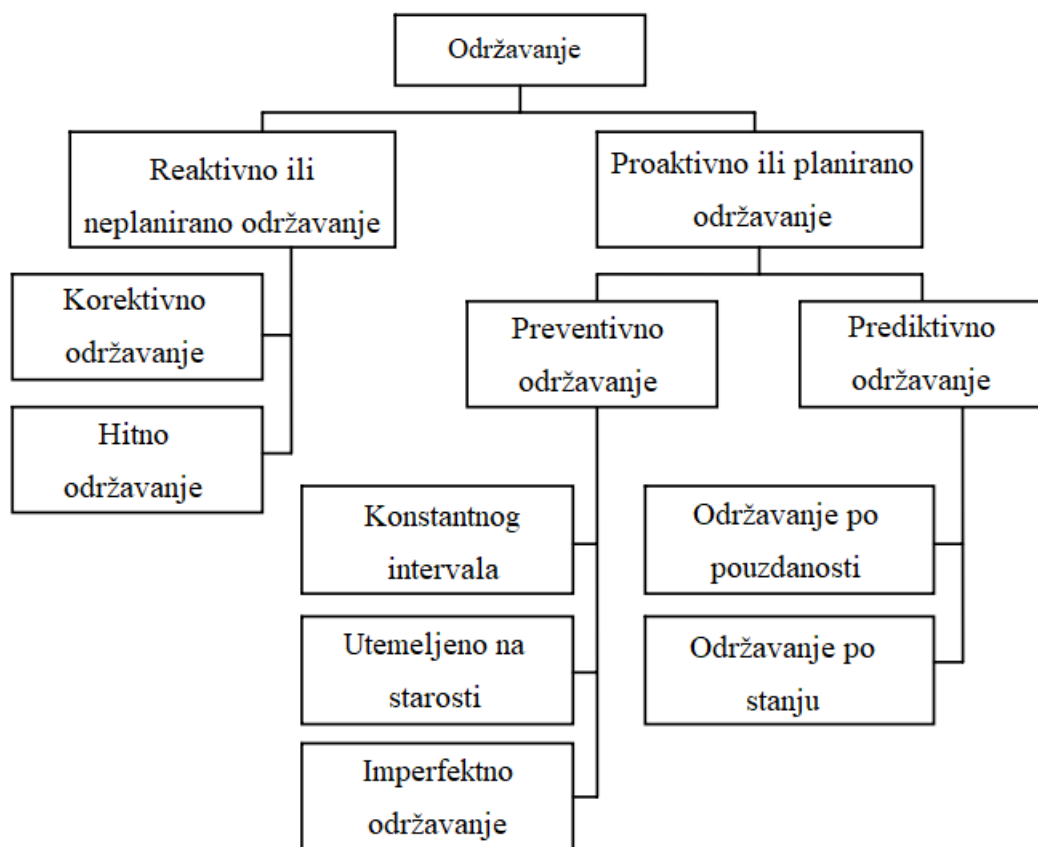
U četvrtom poglavlju opisan je rad eksperimentalnog postava za prikupljanje vibracija s rotirajuće opreme pomoću kojeg su vibracije snimljene na računalo te potom primijenjene na algoritam konvolucijskih neuronskih mreža, posebnog algoritma dubokog učenja za automatsko izvlačenje značajki. Rezultati su naposljetku predstavljeni u petom poglavlju.

## 2. ODRŽAVANJE STROJNE OPREME

Održavanje opreme unutar industrijskih postrojenja je jedna od najvažnijih stvari za uspostavljanje ispravne, funkcionalne i profitabilne industrijske cjeline. Zanemarivanje odjela održavanja i uskraćivanje potrebnih financija za operacije održavanja može višestruko naštetiti cjelokupnom postrojenju. Troškovi održavanja čine veliku većinu operativnih troškova svih proizvodnih pogona. Zavisno od specifične industrije, troškovi održavanja mogu dosežati od 15 do 60 % cjelokupnog profita [1]. Troškovi održavanja se mogu višeznačno protumačiti. Istraživanja pokazuju da se u SAD-u na svaki dolar potrošen na održavanje strojeva, čak 33 centa potroše na nepotrebne ili nepropisno urađene operacije održavanja [1]. S obzirom na to da se u SAD-u godišnje potroši više od 200 milijardi dolara na održavanje, trošak od preko 60 milijardi dolara uloženi na nepotrebne operacije predstavlja ozbiljan problem [1]. Takav trošak ujedno onemogućava stvaranje kvalitetnih i konkurentnih proizvoda, te na kraju može rezultirati i slomom cjelokupnih industrijskih regija, pa i gospodarstava. Glavni razlog takvog neefikasnog upravljanja održavanjem bio je nedostatak realnih podataka koji bi opravdali stvarnu potrebu za operacije održavanja tvorničkih strojeva, sustava i opreme. Raspored održavanja bio je planiran, a u mnogim slučajevima još uvijek jest, prema statističkim trendovima ili po stvarnim kvarovima strojeva. Naglim razvojem mikroprocesorske industrije te povezanih instrumenata za prikupljanje podataka i računalnih sustava za njihovu obradu, došlo je do prilike za omogućavanje kvalitetnijeg upravljanja i izvođenja operacija održavanja strojeva i opreme. Mogućnosti primjene novih tehnologija imaju potencijal sprječavanja preuranjenih popravaka te posljedično voditi ka smanjenju troškova održavanja

### 2.1. Strategije održavanja

Kako bi se razumjele moderne strategije održavanja, potrebno je prikazati i prijašnje te se tako može dobiti i uvid u motivaciju za razvoj novih strategija. Strategije održavanja se u osnovi mogu podijeliti na reaktivne i proaktivne strategije. Općenita taksonomija i podjela strategije odnosno filozofija održavanja prikazana je grafički na Slici 1.



Slika 1 Taksonomija strategija održavanja [2]

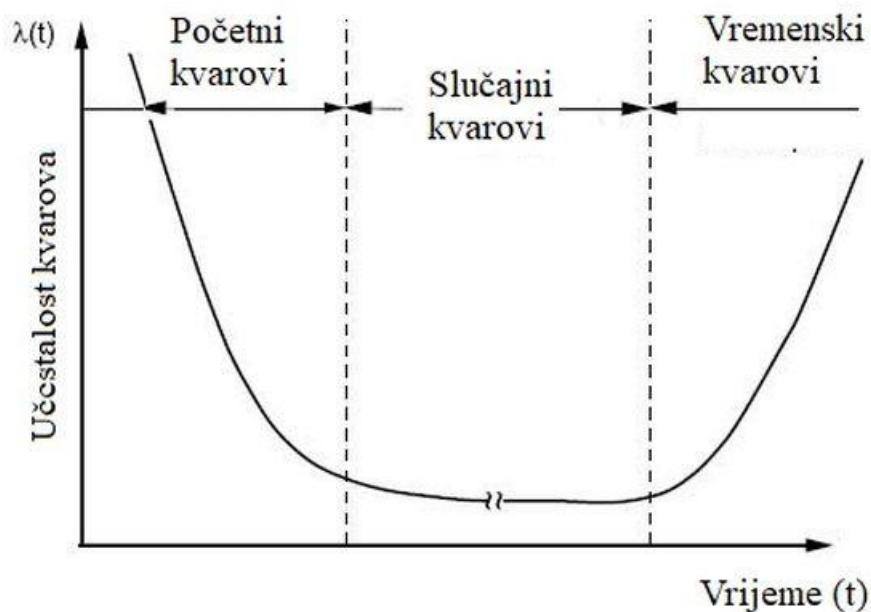
### 2.1.1. Korektivno održavanje

Logika korektivnog održavanja ili radi-do-kvara strategije (eng. *Corrective maintenance, Run-to-Failure*) je prilično jednostavna. Kada se oprema pokvari, onda je treba popravljati, a dok funkcionira ne treba je dirati. Takva strategija podrazumijeva najmanju brigu o održavanju sustava te najčešće predstavlja najskuplju strategiju održavanja. Glavni troškovi povezani s ovom strategijom održavanja su troškovi skladišta rezervnih dijelova, prekovremeni troškovi radnika, dugotrajno vrijeme zaustavljanja pogona te niska produktivnost postrojenja. S obzirom da nisu ostvareni nikakvi pokušaji anticipacije kvara, postrojenje koje koristi korektivno održavanje mora biti spremno na sve vrste kvarova svakog dijela opreme. Ovakva reaktivna strategija održavanja uzrokuje visoke troškove i smanjenu dostupnost strojeva. Korektivna strategija održavanja uzrokuje u prosjeku tri puta veće troškove održavanja od planiranih i predviđenih popravaka [1].

### 2.1.2. Preventivno - plansko održavanje

Iz prethodnog poglavlja može se iščitati koliko je korektivno održavanje zapravo skupa strategija održavanja. Zbog toga se počinje primjenjivati preventivna strategija održavanja, s

unaprijed vremenskim planiranim operacijama održavanja. Drugim riječima, operacije održavanja su planirane na statističkim značajkama dosadašnjih kvarova dijelova i strojeva. Održavanje temeljeno na konstantnom vremenskom intervalu rada opreme se obavlja prema unaprijed utvrđenim vremenskim intervalima koji su izabrani tako da balansiraju visoke troškove održavanja i opasnost od kvara. Na Slika 2 prikazan je tipičan statistički životni vijek trajanja jednog dijela opreme koji je karakterističan za elektromehaničke dijelove.



**Slika 2 Krivulja „kade“ – dijagram učestalosti i vrsti kvarova**

Dijagram učestalosti kvarova pokazuje da stroj ima visoku vjerojatnost kvara nakon instalacije, potom ima dulji vijek trajanja s malom vjerojatnošću za kvar, dok se ta vjerojatnost povećava nakon određenog vremena eksploatacije. U preventivnom održavanju, popravci su planirani pomoću sličnih statističkih krivulja. Glavni izazov kod ovakve strategije je što takve statističke značajke nisu jednake za sve dijelove opreme u svim uvjetima rada. Na primjer, radni vijek trajanja centrifugalne pumpe može biti statistički određen na 18 mjeseci, pa će se pomoću preventivne strategije održavanja pumpa zamijeniti ili popraviti nakon 17 mjeseci rada, čime će se spriječiti moguća havarija ili veći kvar. No, ista pumpa u drugim uvjetima rada može imati kraći ili dulji vijek trajanja. U slučaju preuranjene akcije održavanja stvara se nepotrebn trošak zamjene zdravog strojnog dijela, dok se u slučaju zakašnjele reakcije dolazi do korektivne akcije koja može uzrokovati veće troškove popravka.

Preventivno plansko održavanje je primjenjivo s opremom za koju se relativno lagano i sigurno može ustvrditi predviđeni vijek trajanja, kao što je oprema koja često radi u kontroliranim uvjetima rada. Neke komponente uistinu pokazuju znakove trošenja i disfunkcije u pravilnim

vremenskim intervalima, ali s većinom bitnih dijelova kao što su kuglični valjni ležajevi, postoji velika statistička varijacija oko srednje vrijednosti predviđenog vijeka trajanja, te se strategija preventivnog planskog održavanja teško može primijeniti bez nepotrebnih troškova. Jedini način da se minimiziraju troškovi održavanja i popravka, kao i mogućnost kvarova, jest taj da se konstantno prati zdravstveno stanje strojne opreme, te da se na temelju sadašnjeg i prijašnjeg stanja stroja daju predikcije kvarova i njegovog životnog vijeka.

### **2.1.3. Prediktivno održavanje**

Prediktivno održavanje (*eng. Predictive Maintenance – PdM*) je jedna od najnovijih strategija održavanja koju implementiraju razni sektori industrije, pogotovo oni u kojima je pouzdanost sustava ključna, kao što su hidro, termo i nuklearne elektrane, transportni i komunikacijski sustavi. U osnovi, prediktivno održavanje primjećuje pogoršavanje komponenti sustava i predviđa kvarove i moguće prestanke rada opreme. Zadatak takve strategije je detektiranje ranih stadija kvara opreme i iniciranje akcije održavanja u pravo vrijeme. U industrijski integriranom obliku, strategija prediktivnog održavanja podrazumijeva:

- prikupljanje podataka o stanju sustava ponajviše preko senzora ili bežičnih mrežnih sustava
- procjena prikupljenih podataka
- određivanje prikladnog vremena intervencije
- odlučivanje i provođenje potrebnih akcija održavanja, kao što su izdavanje radnih naloga, aktiviranje osoblja održavanja, naručivanje rezervnih dijelova...

Glavna razlika između preventivnog i prediktivnog održavanja je u planiranom vremenu obavljanja operacija održavanja. U prvom slučaju intervali su fiksno unaprijed određeni dok se u potonjem intervali održavanja adaptivno mijenjaju i prilagođavaju. Na primjer, promjena ulja u automobilu je većinom fiksirana na periode od 20 000 – 30 000 prijeđenih kilometara, što je ekvivalent preventivnog održavanja. Kada bi se povremenim pregledima stanja ulja utvrđivala njegova kakvoća i svojstva podmazivanja, vijek trajanja između promjena ulja bi se mogao značajno produžiti, što je glavna prednost prediktivnog održavanja. Ostale prednosti prediktivnog održavanja su brojne:

- Povećanje operativnog životnog vijeka trajanja komponenti i opreme
- Smanjenje vremena nedostupnosti opreme
- Smanjenje cijene dijelova i troškovi radne snage



- Veća kvaliteta proizvoda
- Veći moral zaposlenika
- Uštede u potrošnji energije
- Procjena od 8% do 12% uštede u odnosu na preventivno održavanje [4]

Nedostaci prediktivnog održavanja su uglavnom vezani uz početne troškove implementacije:

- Povećani troškovi investiranja u opremu
- Povećani troškovi investicija u obuku radnika
- Potencijali uštede nisu jasno vidljivi menadžmentu

Ipak da se navedeni nedostaci isplate premostiti i da će se takve investicije s vremenom uvelike isplatiti pokazuje niz istraživanja koja su dala sljedeće podatke [4]:

- Smanjeni ukupni troškovi održavanja: 25% – 30%
- Eliminacija totalnih kvarova: 70% - 75%
- Redukcija vremena zastoja: 35% - 45%
- Povećanje produktivnosti: 20% - 25%

## 2.2. Održavanje po stanju

Prediktivno održavanje po stanju podrazumijeva donošenje odluka o operacijama održavanja zasnovane na promatranju „stanja“ sustava i njegovih komponenti. Ideja u ovom pristupu je predviđanje vjerojatnosti kvara na dijelu promatrane opreme, te korištenje te informacije za donošenje daljnjih odluka o operacijama održavanja. Nadalje, ovakav pristup zahtjeva pouzdane informacije o stanju sustava i sofisticirane tehnologije za obradu prikupljenog signala [5]. Stanje nekog sustava određeno je parametrima koji se kontinuirano prate i koji su specifični za sustav ili dio sustava. Prednost ove strategije je očita jer se odluke donose isključivo temeljene na stvarnim, fizičkim podacima koji na najbolji način reflektiraju stanje strojne opreme. Najčešće mjerne tehnike za praćenje stanja opreme su:

- Mjerenje i analiza vibracija
- Termografija
- Ultrazvuk
- Akustična emisija
- Praćenje oštećenja i krhotina

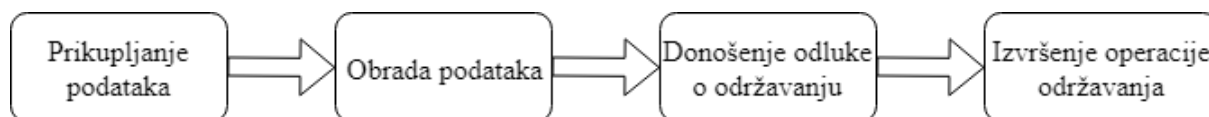
- Analiza maziva
- Mjerenje procesnih veličina (tlak, temperatura...)
- Vizualna inspekcija i ostale neinvazivne tehnike

Pretpostavka iza filozofije prediktivnog održavanja je da će konstantno praćenje stanja stvarnih mehaničkih uvjeta rada sustava:

- osigurati maksimalno vrijeme između popravaka
- minimizirati broj i trošak nepredviđenih zahvata na opremi
- povećati ukupnu dostupnost opreme i produktivnost postrojenja

Ako se propisno i efikasno implementira, strategija održavanja po stanju može značajno smanjiti troškove smanjujući nepotrebno planirane operacije popravaka i zamjene opreme [6]. Da bi se održavanje po stanju pravilno sprovelo, potrebno je pratiti glavne korake tijekom implementacije [6]:

1. Prikupljanje podataka (informacija), kako bi se opskrbili relevantnim podacima o zdravstvenom stanju stroja.
2. Procesuiranje podataka u cilju obrade i analize podataka skupljenih u prvom koraku za bolje razumijevanje i interpretaciju.
3. Donošenje odluke o postupku održavanja i preporuke efikasne operacije



Slika 3 Tipični koraci pri održavanju po stanju [6]

### 2.3. Mjerenje i analiza vibracija

Analiza vibracija strojnih dijelova je najčešća tehnika pomoću koje se implementira prediktivno održavanje po stanju unutar industrijskih sustava. Koristi se ponajviše na rotirajućim strojnim dijelovima ili uređajima koji imaju ciklička ponavljanja (pumpe, kompresori...) te na zupčanicima ili rotirajućim električnim uređajima. Prvi znakovi kvarova mogu se vibracijskom analizom uočiti jako rano, čak i mjesecima prije stvarne pojave kvara. Stanje opreme se praćenjem vibracija može kontrolirati konstantno, a rezultati obrade informacija dolaze u realnom vremenu. Mnoge efikasne tehnike obrade signala mogu biti primijenjene na vibracije te iz njih očitati slabe indikacije mogućeg kvara koje su često zamaskirane šumom i ostalim nebitnim informacijama.

Praćenje vibracija na strojnim dijelovima može biti podijeljeno na *konstantno* i *povremeno* mjerenje [5]. Kritični strojni dijelovi često imaju trajno pričvršćene elektroničke pretvarače signala koji konstantno prate rad stroja kako bi strojevi mogli biti jako brzo ugašeni u slučaju naglih promjena u radu koje bi mogle indicirati katastrofalne kvarove. Iako će automatsko gašenje postrojenja svakako uzrokovati gubitke u proizvodnji, zastoji i troškovi zbog pojave stvarnih kvarova su neusporedivo većeg razmjera, te mogu ozbiljno narušiti funkcionalnost stroja. Prednosti konstantnog praćenja stanja stroja su:

- Brzo reagiranje na nagle promjene u radu i najveći potencijal u zaštiti i sigurnosti kritične komponenti.
- Najbolji način zaštite od pojave naglih i nepredvidljivih kvarova, kao što su pojave disbalansa ili iskrivljenja osi vrtnje na vratilima mehaničkih strojeva.

Nedostaci konstantnog praćenja su:

- Cijena trajnog ugrađivanja pretvarača na strojne dijelove je velika i može se primijeniti samo na kritične komponente
- Obzirom da reakcija na pojavu anomalije mora biti brza, analiza konstantnog praćenja vibracija svodi se na praćenje jednostavnih vibracijskih parametara (RMS, vršne vrijednosti...)

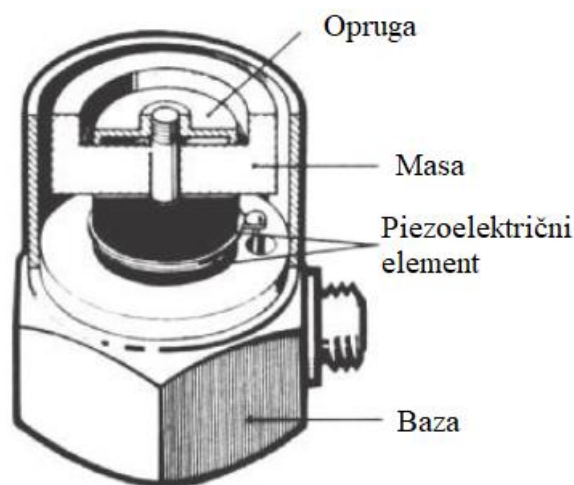
Iako se kombiniranjem povremenog praćenja stanja u pouzdano određenim intervalima mogu postići dobri rezultati u održavanju po stanju, postoji potreba za implementiranjem integriranih sustava prediktivnog održavanja koji će konstantno vršiti monitoring ključnih komponenti sustava i pravovremeno upozoravati na pojavu potencijalno opasnih defekata.

Vibracijska analiza je posebno korisna kod uočavanja i rješavanja sljedećih vrsta kvarova kod rotirajuće i cikličke opreme:

- Imbalans / Debalans. Neusklađenost centra mase poprečnog presjeka osovine s centrom rotacije.
- Ekscentričnost.
- Nesuosnost spojki i ležajeva
- Rezonancijski problemi
- Nezategnutost mehaničkih dijelova
- Iskrivljenje osovine
- Pukotine na osovini

- Istrošeni ili oštećeni zupčanici i ležajevi
- Oštećeni ili pogrešno postavljeni lanci i pojasevi pogonskih dijelova
- Oštećenja na lopaticama turbina

Za mjerenje vibracija stroja najčešće se koriste senzori akcelerometri. Akcelerometri su pretvarači koji daju signal proporcionalan akceleraciji. Najčešće korišteni akcelerometri u praćenju zdravstvenog stanja stroja su piezoelektrični akcelerometri koji koriste specifična piezoelektrična svojstva određenih kristala i keramika. Tipičan dizajn piezoelektričnog akcelerometra je tzv. „kompresijski“ tip senzora, prikazan na Slika 4.



**Slika 4 Unutrašnjost tipičnog piezoelektričnog akcelerometra [5]**

Piezoelektrični element je postavljen između mase i baze, a cijeli uređaj u kompresiji drži opruga. Kada je baza senzora pričvršćena na vibracijski objekt, masa je primorana da prati kretanje baze preko piezoelektričnog elementa koji se ponaša kao kruta opruga. Varirajuće inercijske sile mase utječu na piezoelektrični element tako što ga blago deformiraju, stvarajući pomak koji je proporcionalan promjeni akceleracije. Piezoelektrični element pod deformacijom proizvodi električni signal koji je proporcionalan akceleraciji, a mjerljiv je u  $\frac{pC}{ms^2}$  (pikoKuloni po metru sekundi na kvadrat). Taj signal se pretvara u napon i pojačalom pojačava do potrebne mjere.

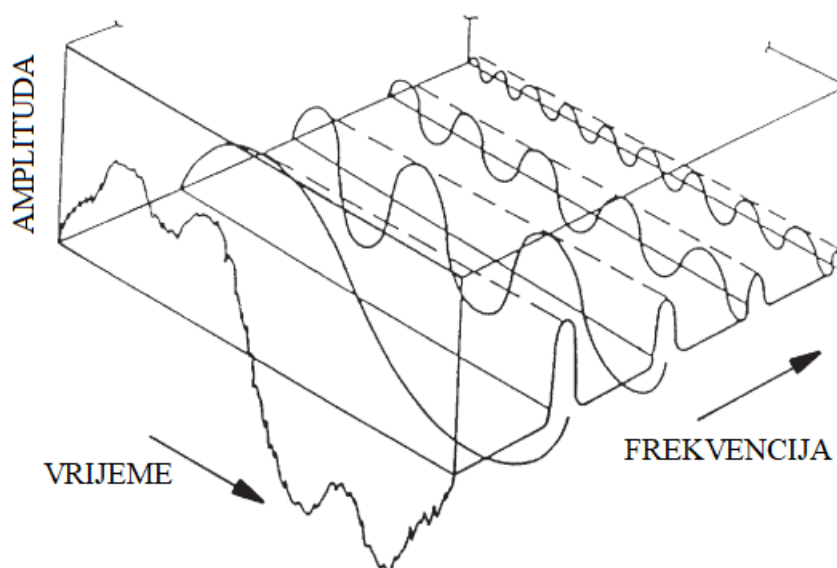
Vibracijski profili mogu biti prikupljeni i prikazani u dva različita oblika:

- Vremenska domena
- Frekvencijska domena

Signal u vremenskoj domeni je prikazan kao amplituda na vertikalnoj osi i vrijeme na horizontalnoj osi. Veličina horizontalne osi ovisi o frekvenciji odnosno vremenu uzorkovanja

tijekom mjerenja signala. Profil vibracija u vremenskoj domeni je zbroj svih vibracijskih komponenti koji su prisutni na opremi tijekom mjerenja. Kod složenih vibracijskih signala, gotovo je nemoguće uočiti sve frekvencije od kojih je signal sastavljen.

Zbog toga se u analizi vibracijskog signala najčešće koristi signal prikazan u frekvencijskoj domeni gdje je vremenski trag signala rastavljen na diskretne frekvencijske komponente koristeći matematički proces zvan „Fourierova transformacija“ ili „brza Fourierova transformacija“ (eng. *Fast Fourier Transformation – FFT*). U frekvencijskoj domeni, na vertikalnoj osi je amplituda, ali na horizontalnoj osi označene su frekvencije vibracija. Prema teoriji Fourierove serije, svaki periodični signal može se predstaviti sumom pravilnih sinus valova, a isto pravilo vrijedi i za kompleksne vibracijske signale. Pomoću spektralne analize, može se ustanoviti na kojim frekvencijama vibriranja se pojavljuju veće amplitude te time pronaći određene karakteristike čestih kvarova opreme.



**Slika 5 Odnos između vremenske domene i frekvencijske domene [1]**

Preventivno održavanje analizom vibracija temelji se na dvije osnovne činjenice:

1. Amplituda svake pojedine vibracijske komponente će ostati konstantna ukoliko dinamika tehničkog sustava ostane nepromijenjena
2. Svaki tip kvara ima karakterističnu komponentu vibracijskih frekvencija koje se mogu filtrirati i definirati

Na osnovu navedenih činjenica moguće je implementirati različite strategije i tehnike primjene preventivnog održavanja pomoću mjerenja i analize vibracije industrijske opreme.

## 2.4. „Internet stvari“ – primjena u industriji

„Internet stvari“ (*eng. Internet-of-Things*) ili „Internet svega“ je termin koji podrazumijeva novu tehnološku paradigmu predstavljenu kao mrežu međusobno povezanih različitih fizičkih objekata: uređaja, strojeva, vozila, građevina koji su integrirani pomoću elektronike, softvera, senzora i mrežne povezanosti – što omogućuje tim objektima da prikupljaju i razmjenjuju podatke. Internet stvari je prepoznat kao jedna od najvažnijih grana razvoja tehnologije u budućnosti i već dobiva na značaju u svijetu industrije i poduzetništva. [7]

Iako je automatizacija u modernoj industriji u kojoj prednjači autoindustrija uzela zamaha već u zadnjih 30 godina, ono što takve automatizirane pogone čini još efikasnijima je međusobna povezanost svih radnih jedinica, koju može kontrolirati jedan glavni informacijski centar. Takva povezanost može dodatno smanjiti gubitke uzrokovane ljudskim faktorima, dok se ljudska radna snaga može fokusirati na razvijanje autonomnih sustava proizvodnje i održavanja. Povezanost kibernetičko-fizičkih uređaja unutar postrojenja naziva se „Industrijski Internet stvari“ (*eng. Industrial Internet-of-Things*) ili jednostavnije „Industrijski Internet“ (*eng. Industrial Internet*).

Arhitektura tehnologije Internet stvari može se predstaviti u četiri različita sloja:

U *senzorskom sloju* bežični inteligentni sustavi sposobni su automatski prikupljati i razmjenjivati podatke između različitih uređaja. Unutar nekih industrijskih sektora svakom povezanom uređaju dan je posebni univerzalni identifikator (*eng. UUID – Universal Unique Identifier*) te se pomoću UUID-a svaki uređaj može pratiti, identificirati i na njemu izvršiti potrebnu intervenciju. [8]

Sloj	Opis
Senzorski sloj	Senzorski sloj integriran je s postojećim hardverom (RFID, senzori, aktuatori...) za prikupljanje potrebnih podataka
Mrežni sloj	Ovaj sloj pruža osnovnu mrežnu infrastrukturu i prijenos podataka žičnom ili bežičnom mrežom
Servisni sloj	Sloj za pružanje servisne podrške korisniku pomoću posredničkog softvera
Sučelje	Sloj sa sučeljem pruža korisniku mogućnost interakcije i korištenje različitih dostupnih aplikacija

**Tablica 1 Četiri sloja arhitekture tehnologije Internet stvari [8]**

Zadaća *mrežnog sloja* je povezivanje svih komponenti sustava i omogućavanje dijeljenja informacija unutar sustava. Unutar mrežnog sloja potrebno je pobrinuti se o stvarima kao što su: tehnologija mrežne povezanosti (žična, bežična...), energetska efikasnost mreže, kvaliteta usluge, obrada podataka i signala, sigurnost prijenosa podataka i očuvanje privatnosti.

*Servisni sloj* oslanja se na tehnologiju posredničkih softvera (*eng. middleware*) koja pruža funkcionalnost sustavu i integrira usluge i aplikacije u Internetu stvari. Posrednički softver omogućava efikasnu platformu za servisnu podršku sustava i omogućava integriranje svih pripadajućih softvera i hardvera u strukturi Interneta stvari.

Unutar Interneta stvari, poglavito u industrijskom postrojenju, može biti uključeno mnogo uređaja od različitih proizvođača te može se dogoditi da ne slijede jednake komunikacijske protokole. Zbog te heterogenosti uređaja može doći do problema s razmjenom informacija, komunikacijom između uređaja i donošenjem odluka. Zbog toga je potrebno efikasno *sučelje* ili *sučeljni sloj* koji će moći podržati interakciju svih uređaja spojenih u Internet stvari i korisnika odnosno onoga tko upravlja cijelim sustavom. Dobrim i efikasnim sučeljem smatra se implementacija univerzalnog „Uključi i radi“ (*eng. Plug & Play*) sustava koji omogućava jednostavno uključivanje, isključivanje i kontroliranje objekata u sustavu. [8]

## 2.5. Strategija Industrija & Održavanje 4.0

Prediktivno održavanje posljednjih godina doživljava proboj u primjeni u svim svjetskim, modernim industrijskim pogonima. Razlog za takav revolucionaran pomak u svijesti industrijskog upravljačkog kadra je dostupnost velikim količinama podataka, te mogućnost spremanja i obrade masovnih podataka koji dolaze s mnoštva ugrađenih senzora. Prediktivno održavanje ipak nije nova pojava u održavanju, ali do danas se ta strategija razvijala u tri koraka prema [9]:

1. Vizualna inspekcija – periodična vizualna inspekcija opreme, odluke se donose isključivo temelju znanja i stručnosti radnika.
2. Instrumentalne inspekcije – periodične inspekcije opreme pomoću specijaliziranih instrumenata, odluke se donose na temelju znanja radnika i očitavanja sa instrumenata.
3. Monitoring u stvarnom vremenu – kontinuirano praćenje stanja opreme pomoću senzora, odluke o održavanju donose se prema unaprijed utvrđenim pravilima ili kritičnim razinama.

Pristup velikim podacima (*eng. Big Data*) i napredak analitičkih postupaka označava početak četvrtog koraka u razvoju prediktivnog održavanja:

4. PdM 4.0 – kontinuirano praćenje stanja opreme pomoću senzora, uzbune o pojavi kvara donose napredni analitički algoritmi s mogućnostima predikcije, kao što su algoritmi strojnog učenja.

Povijesno gledano, industrijska proizvodnja se u ljudskoj civilizaciji razvija kroz zadnja dva stoljeća, a najveći napreci su se događali prilikom otkrića i inovacija koji su u potpunosti mijenjali smjer razmišljanja i funkcioniranja tadašnjih industrijskih postrojenja. Takvi zaokreti nazivaju se *industrijske revolucije*, a dosada su zabilježene tri takva događaja u povijesti industrijskog razvoja. Razvoj industrijske proizvodnje može se povezati sa razvojem strategija održavanja, koje su nastale kao posljedica potreba za efikasnijom upotrebom jednog od najvažnijih proizvodnih resursa – proizvodne opreme. Tako se održavanje razvijalo uz industrijske revolucije:

- *Industrija 1.0 : Reaktivno održavanje.* 19. i početak 20. stoljeća. Na održavanje se nisu trošili dostupni resursi, te se prakticirala strategija rada stroja do kvara. Bez ikakvih troškova održavanja i radne snage koja se time bavila, vrijeme nedostupnosti strojeva je bilo veliko.



- *Industrija 2.0 : Preventivno održavanje.* Sredina 20. stoljeća. Periodično održavanje je uključivalo mijenjanje komponenti prije pojave kvara, a intervali održavanja su određivani iskustveno ili na preporuku proizvođača opreme.
- *Industrija 3.0 : Prediktivno održavanje.* Krajem 20. i početkom 21. stoljeća. Počinje implementacija održavanja po stanju u realnom vremenu, praćenjem stanja opreme u operaciji. Granice poslije kojih se proglašava vrijeme za održavanje su određene statističkim i iskustvenim mjerilima.

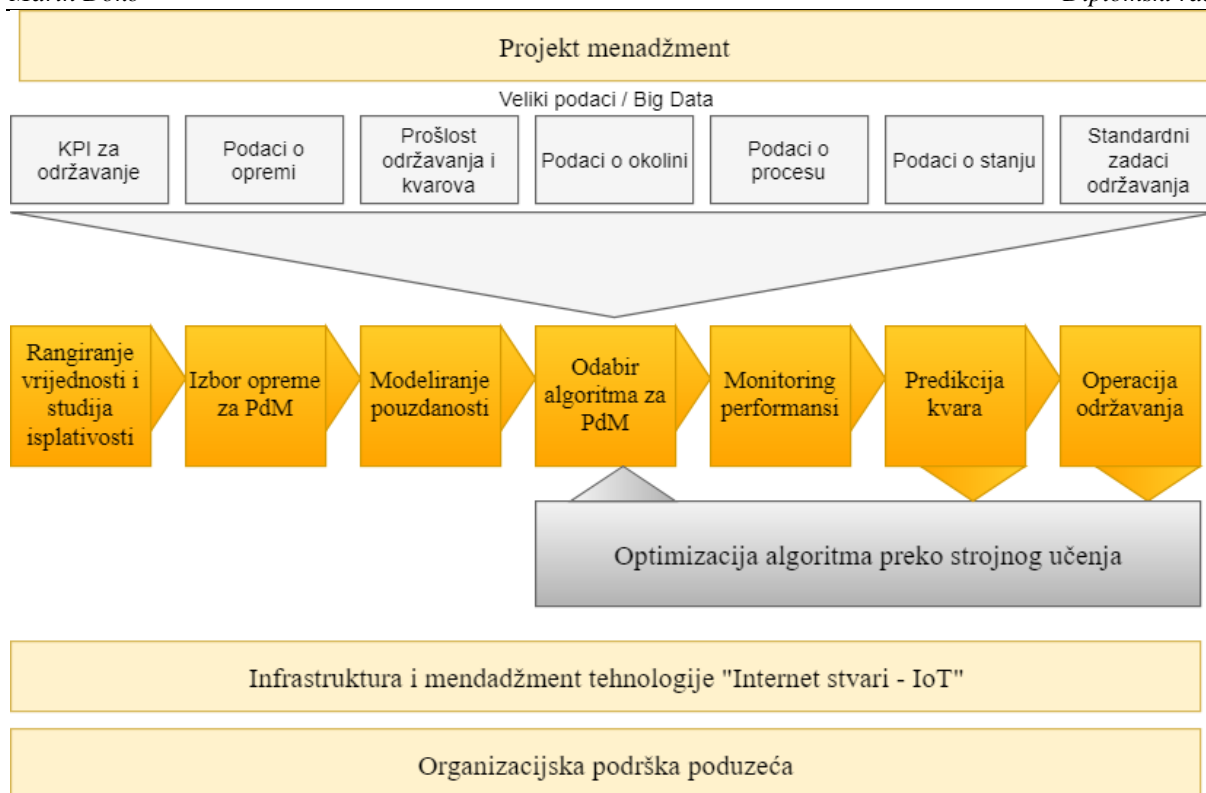
Moderni industrijski stručnjaci i znanstvenici slažu se u međusobnim istraživanjima da je u tijeku začetak četvrte industrijske revolucije. Kao što je opisano u [10], četvrta industrijska revolucija promijenit će dizajn, manufakturu, operacije i servis proizvoda i proizvodnih sustava. Implementacija Industrije 4.0 u gospodarstvo se započela provoditi u Njemačkoj, gdje se procjenjuje da će u idućih 10 godina doprinijeti godišnje s 1% ukupnog BDP-a, stvoriti 390 000 novih radnih mjesta, te dodati €250 milijardi u proizvodne investicije. Četvrta industrijska revolucija potaknuta je ubrzanim tehnološkim razvojem i otkrićima, a ona koja se najviše ističu su: Kibernetičko-fizički sustavi (*eng. Cyber-physical systems – CPS*), analitika velike količine podataka (*eng. Big data analytics*), računanje u oblaku (*eng. Cloud computing*), te Internet stvari (*eng. Internet-of-Things*) [11]. Razvojem navedenih tehnologija prediktivno održavanje postaje jedan od glavnih ciljeva Industrije 4.0 i jedna od centralnih točki stvaranja prednosti nad konkurencijom u svijetu industrije. Tako se gornjem popisu industrijskih revolucija i povezanih strategija održavanja može dodati i sljedeća, aktualna točka razvoja tog polja:

- *Industrija 4.0: Prediktivno održavanje 4.0 – PdM 4.0:* Kibernetičko-fizički sustavi u postrojenju međusobno su povezani bežičnim senzorskim mrežama. Povećan pristup velikoj količini podataka omogućuje bolji uvid u stanje sustava, a primjena umjetne inteligencije i strojnog učenja omogućuje predviđanje dosad nepredvidljivih kvarova.

Implementacija PdM 4.0 sustava unutar poduzeća predstavlja izazovan zadatak kojeg većinom nije lako provesti. Istraživanje provedeno u [9] na osnovu podataka prikupljenih iz različitih poduzeća daje smjernice za implementaciju integriranog programa PdM 4.0:

1. *Rangiranje vrijednosti i značaja opreme i studija isplativosti.* Potrebno je utvrditi za koju opremu je isplativo uvesti strategiju PdM 4.0 kako bi joj povećali pouzdanost i radni vijek. Mogući kandidati su samo visoko ili srednje kritična oprema i oprema za koju se može prikupiti velika količina podataka.
2. *Izbor opreme na koju će se primijeniti PdM 4.0.* Potrebno je na osnovu prethodnih studija izabrati dio opreme za provedbu strategije.

3. *Modeliranje pouzdanosti.* Korištenjem analize uzroka korijena kvara (RCA) i analize utjecaja i posljedica pogrešaka (FMEA) utvrditi smjer implementacije PdM 4.0. Otkriti koji tip podataka skupljati i koje tipove kvarova istraživati.
4. *Odabir algoritma prediktivnog održavanja.* Izbor prikladnog algoritma je ključna odluka u određivanju efikasnosti modela prediktivnog održavanja. Ovdje dolaze do izražaja samo-učeći algoritmi strojnog učenja koji stvaraju smislene uvide iz generiranih skupova podataka.
5. *Monitoring performansi u stvarnom vremenu.* Algoritam procesira podatke iz raznih izvora: senzora na izabranoj opremi, senzora okoline opreme, baze podataka prošlog održavanja i kvarova opreme; kako bi vizualizirao performanse u stvarnom vremenu.
6. *Predikcija kvara opreme (rano upozorenje).* Algoritam počinje predviđati buduće kvarove. Zaustavljanje prividno zdravog dijela opreme može u početku zahtijevati veliko povjerenje rukovodećih osoba, posebno ako je menadžment ili osoblje održavanja skeptično prema analizi podataka. PdM 4.0 može se u tom slučaju za početak provesti paralelno s postojećim programom održavanja, te uspješnom predikcijom podići povjerenje u buduće rezultate.
7. *Propisivanje operacije održavanja.* Krajnji nivo PdM 4.0 strategije podrazumijeva da algoritam, ne samo što radi predikcije vremena kvara, nego i pregledava standardne postupke održavanja iz baze podataka i propisuje najbolju proceduru za izbjegavanje mogućeg kvara. Algoritam može također i sam izvršiti taj postupak, propisujući ovlaštenu radni nalog osoblju održavanja.



**Slika 6** Dijagram općenitog pristupa integriranoj implementaciji Prediktivnog Održavanja 4.0. Adaptirano iz [9]

Na Slika 6 prikazan je dijagram preporučenog pristupa strategiji Održavanja 4.0. Uz već spomenute korake implementacije sustava, prikazani su i mogući tipovi velikih podataka potrebnih za ispravno funkcioniranje algoritma prediktivnog održavanja. Osim velike količine podataka, potrebna je i velika brzina prijenosa i obrade podataka. Za provedbu PdM strategije potrebna je infrastruktura „Internet stvari“, koja će najčešće bežično povezati strojnu opremu s podatkovnim centrom održavanja i koja će omogućiti prikupljanje i distribuciju senzorskih podataka. Efikasna IoT infrastruktura podrazumijeva odabir pravih protokola za bežičnu komunikaciju, enkripcija podataka i sigurnost prijenosa podataka.

Najveći izazov pri implementaciji strategije PdM 4.0. trenutno nije odabir prave tehnologije, algoritma ili podataka već pristup unutar organizacije i nedostatak potrebnih digitalnih vještina i digitalne kulture. Uspjeh primjene ove strategije najčešće neće ovisiti o sensorima i algoritmima već o širokom spektru ljudsko-povezanih faktora.

### 3. STROJNO UČENJE

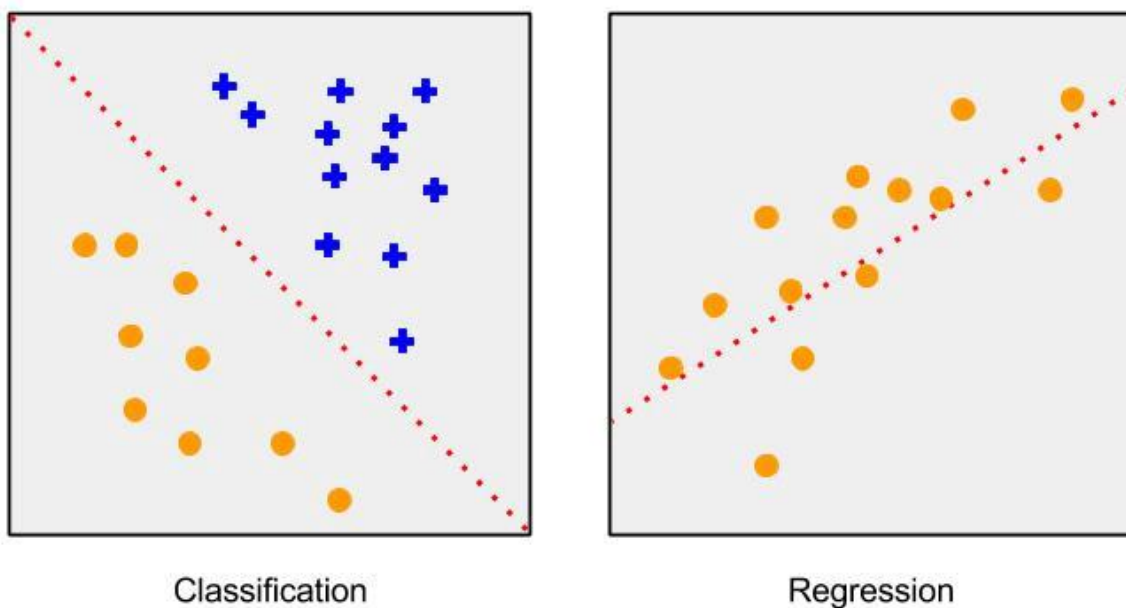
Strojno učenje, jednostavno definirano, omogućuje računalnom programu / aplikaciji da uči iz danih podataka umjesto da bude eksplicitno programirano. Problem strojnog učenja može biti definiran kao [12]:

„Za računalni program se može reći da uči iz *iskustva*  $E$  za neki *zadatak*  $T$  i s nekom mjerom *učinkovitosti*  $P$ , ako se učinkovitost na zadatku  $T$ , mjerena učinkovitošću  $P$ , poboljšava s iskustvom  $E$ “

Iz te definicije mogu se predstaviti današnji moderni zadaci strojnog učenja. Sam proces učenja ne može se smatrat zadatkom strojnog učenja, već samo sredstvom pomoću kojeg se dolazi do rješavanja određenih problema i zahtjeva pri obradi podataka. Zadaci strojnog učenja su opisani kao načini na koji sustavi strojnog učenja procesuiraju primjere, dok su primjeri kolekcije značajki, kvantitativno izmjerenih s nekog objekta u svrhu primjene strojnog učenja. Zadaci strojnog učenja koji se do danas najviše primjenjuju u praksi su [13]:

- **Klasifikacija.** Kod klasifikacijskih zadataka računalni program treba odrediti kojoj od  $k$  klasa pripada neki određeni ulaz u algoritam. Kako bi riješio ovaj problem računalni program mora naučiti funkciju  $f: \mathbb{R}^n \rightarrow \{1, \dots, k\}$ . Kada je  $y = f(x)$ , model klasificira ulaz  $x$  u jednu od naučenih klasa  $k$  preko numeričke vrijednosti izlaza funkcije. U drugoj varijanti klasifikacije, izlaz funkcije  $f$  može biti raspodjela vjerojatnosti preko ponuđenih klasa. Jedan od najpoznatijih i primijenjenih primjera klasifikacije je prepoznavanje objekata na slikama, gdje je ulaz u algoritam slika određene veličine i vrijednosti jačine svakog piksela, dok je izlaz numerička vrijednost koja identificira objekt na slici. Prepoznavanje objekata se pokazalo kao učinkovit algoritam pri prepoznavanju obrisa i kontura ljudskih lica na fotografijama, što je dokazano u [14]. Klasifikacija se može koristiti i u dijagnostici strojeva, gdje se na osnovu različitih stanja stroja može klasificirati tip ili vrsta kvara ili drugog abnormalnog stanja.
- **Regresija.** U ovom tipu zadataka, od računalnog programa se očekuje da predvidi neku numeričku vrijednost nakon zadavanja određenih ulaznih vrijednosti. Kako bi riješio takav zadatak, program mora naučiti funkciju oblika  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ . Regresija se od klasifikacije razlikuje samo po vrsti izlaza tražene funkcije. Najučestaliji primjer regresije u praksi je predviđanje kretanje cijena određenih usluga ili proizvoda na tržištu. Vizualni prikaz razlike između regresije i klasifikacije je prikazan na Slika 7, gdje

isprekidana linija (funkcija) kod regresije prati trend značajki, dok kod klasifikacije odvaja različite klase značajki.

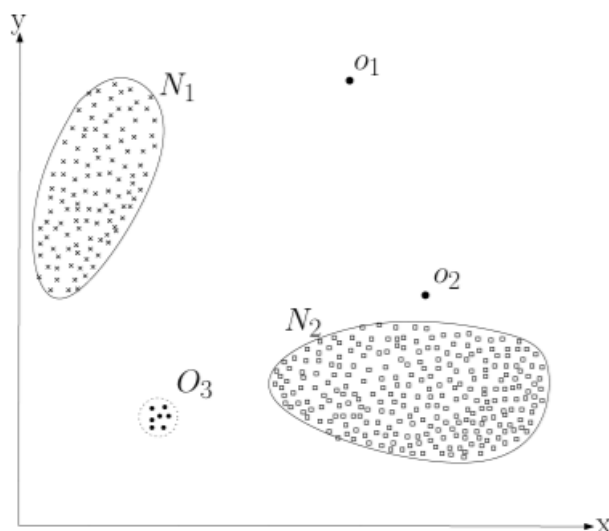


Slika 7 Razlika između klasifikacije i regresije.

- **Transkripcija.** U ovom relativno novom zadatku strojnog učenja od računalnog programa se zahtjeva da promatra relativno nestrukturirani prikaz neke vrste podataka, te da prebaci informacije u diskretni tekstualni oblik. Primjer je optičko prepoznavanje znamenki i slova, gdje se računalu daje kao ulaz slika, dok se kao izlaz očekuje tekst u istom redoslijedu simbola kao na slici. Transkripcija se koristi i kod raspoznavanja govora, gdje je ulaz u algoritam datoteka s informacijama o zvučnom valu koju je potrebno pretvoriti u niz tekstualnih simbola, najčešće slova.
- **Strojno prevođenje.** U zadatku strojnog prevođenja, ulaz u algoritam se već sastoji od niza simbola odnosno riječi na jednom prirodnom jeziku, dok bi izlaz trebao biti prijevod na drugi prirodni jezik. Google Translate je najpoznatiji primjer strojnog prevođenja, a razvoj ovakvih algoritama je tek u začetku.
- **Strukturni izlaz.** Kod zadataka sa strukturnim izlazom, izlaz algoritma strojnog učenja mora biti vektor (ili neki drugi oblik skupa informacija s višestrukim vrijednostima) čiji elementi imaju bitne međusobne odnose. I gore navedena transkripcija i strojno prevođenje, kao i mnogi drugi zadaci strojnog učenja, mogu se svrstati u zadatke sa strukturnim izlazom. Primjer strukturnog izlaza je vektor simbola prepoznatih sa slike

koji zajedno moraju činiti riječ odnosno njihovi elementi u izlaznom vektoru moraju imati određeno značenje i strukturu.

- **Detekcija anomalija.** U ovom tipu zadatka, algoritam mora na osnovu danih značajki naučiti normalno stanje, te potom na novim ulazima detektirati radi li se o anomalijama ili dopuštenim stanjima. Algoritmi detekcije anomalija su potencijalno široko primjenjivi na održavanju strojarških sustava i dijelova, a prostor za primjenu se otvara u medicini, sigurnosnim sustavima ili vojnim sustavima. Primjena ovih algoritama može biti jako značajna jer ranim otkrivanjem anomalije najčešće se sprječava velika i potencijalno opasna šteta. U strojarstvu, anomalija se može otkriti pregledavajući stanje stroja i otkrivanjem početnih stadija kvara. U medicini, detektori anomalija mogu predvidjeti poremećaj rada vitalnih organa, dok u sigurnosnim sustavima mogu očitati krađu kreditnih kartica u Internet kupovini ili pak hakerske napade.



**Slika 8** Primjer rezultata algoritma detekcije anomalije. N1 i N2 su grupirana normalna stanja, dok O stanja označavaju odstupanja od normalnih stanja. [15]

- **Procjena gustoće.** U zadacima procjene gustoće, algoritam strojnog učenja mora naučiti funkciju  $p_{model}: \mathbb{R}^n \rightarrow \mathbb{R}$ , gdje je  $p_{model}(x)$  funkcija gustoće vjerojatnosti (ako je  $x$  kontinuirana vrijednost) ili funkcija mase vjerojatnosti (ako je  $x$  diskretan) na prostoru s kojeg su podaci izvučeni. Da bi ispunio ovaj zadatak, algoritam mora naučiti strukturu viđenih podataka. Algoritam mora naučiti gdje se podaci grupiraju, a gdje je manja šansa da će se budući podaci pojaviti. Ustvari, zadatak strojnog učenja je da implicitno nauči raspodjelu vjerojatnosti danih podataka. Nakon što algoritam dovoljno

dobro nauči raspodjelu vjerojatnosti podataka, ta se vjerojatnost može koristiti za primjenu drugih zadataka strojnog učenja kao što je klasifikacija ili detekcija anomalije.

- **Klasterizacija / grupiranje.** Klasterizacija (eng. *Cluster*) je zadatak strojnog učenja kod kojeg je cilj grupirati određene značajke na takav način da su značajke u istoj skupini sličnije od značajki iz druge skupine. Najpoznatiji algoritmi klasterizacije su grupiranje po k-srednjih vrijednosti (k-means), hijerarhijsko grupiranje i grupiranje po k najbližih susjeda (k-NN).

Zavisno od toga kakva je vrsta podataka odnosno iskustva dostupna tijekom procesa učenja, algoritmi strojnog učenja se najčešće dijele na nadzirano ili nenadzirano učenje, te potaknuto učenje. Tijekom učenja podaci za algoritam se mogu podijeliti na trening, validacijske i testne podatke. Set trening podataka služi za učenje modela u svrhu određene primjene. Set testnih podataka služi kako bi se procijenila efikasnost modela na dosad neviđenim podacima. Validacijski skup podataka može poslužiti kao nepristrani skup podataka za evaluaciju modela, kako bi se što bolje prilagodili hiperparametri i konstrukcija cijelog modela, te kako bi model na testnim podacima dao bolje rezultate. Podaci koji se koriste kao ulaz u algoritme strojnog učenja najčešće se nazivaju značajke (eng. *features*). Značajka se može definirati kao individualna, mjerljiva veličina ili karakteristika promatranog fenomena [16]. Biranje pravih, informativnih, diskriminantnih i neovisnih značajki je krucijalan korak za razvijanje efektivnih algoritama strojnog učenja ili prepoznavanja obrazaca.

- *Nadzirano učenje.* Čest pristup modeliranju algoritama strojnog učenja polazi s pretpostavkom da unaprijed poznajemo labele (oznake, skupove) kojima ulazni podaci pripadaju. Model nadziranog strojnog učenja treniran je korištenjem skupa primjera, gdje svaki primjer ima svoj ulazni vektor i izlaznu vrijednost,  $[X, y]$ . Cilj nadziranog učenja je pronalazak funkcije koja će ispravno mapirati različite ulaze na ciljane izlaze:

$$y = f(X) \tag{1}$$

Primjer: Prikupljanje podataka vibracija sa senzora na ležajevima:

$X = [X_1 X_2 \dots X_n]$  – matrica trening primjera. Svaki  $X$  predstavlja jedno mjerenje.

$y = [y_1 y_2 \dots y_n]$  – Svakom mjerenju pripada određena labela, odnosno karakterizacija kao što je: Signal sa zdravog stroja, signal s oštećenim vanjskim prstenom ležaja, signal s oštećenim unutarnjim prstenom ležaja, nakrivljena osočina...

- *Nenadzirano učenje.* Ovakav pristup modeliranju algoritama strojnog učenja polazi od pretpostavke da podaci nisu označeni, odnosno prethodno se ne poznaje izlazni vektor

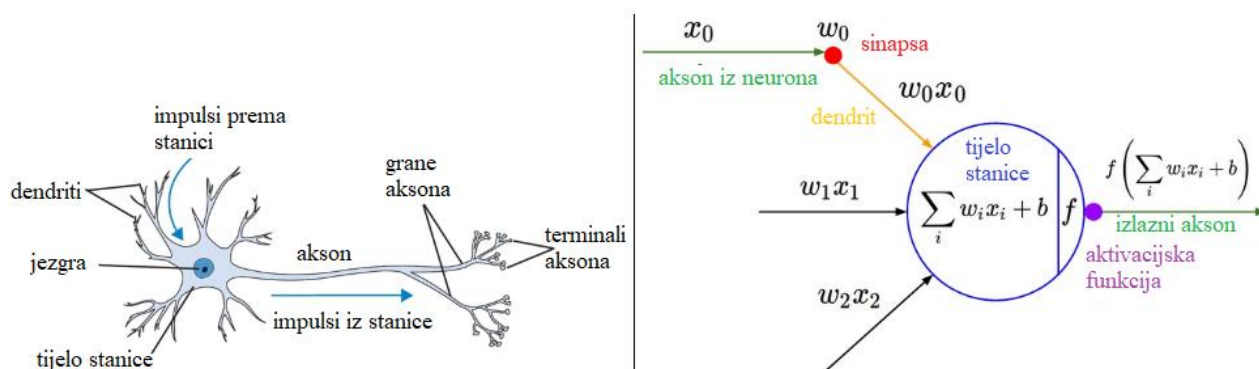
podataka. Cilj nenadziranog učenja je modeliranje strukture ili distribucije podataka kako bi se o tim podacima naučilo više informacija. U nenadziranom učenju nema učitelja i nema „točnog“ odgovora, nego se prepušta algoritmu da sam nauči i prezentira zanimljive strukture u podacima.

Postoji i kombinacija nadziranog i nenadziranog učenja, koja se naziva *polu-nadzirano učenje*. Takav algoritam se primjenjuje kada je dio dostupnih podataka unaprijed označen da pripada jednoj klasi, dok za drugi dio podataka nemamo poznate izlazne vrijednosti. Kod analize vibracija strojeva s rotirajućim dijelovima, najlakši i najdostupniji podaci su oni sa zdravog uređaja, kao što može biti i iščitano s dijagrama na Slika 2. Po tome, primjenjujući takve podatke na algoritme strojnog učenja, većina podataka bi mogla biti označena kao „ispravna“, pa bi s algoritmom polu-nadziranog učenja stvorili potprostor u koji bi bile svrstane značajke ispravnog signala, dok bi značajke van tog signala značile odstupanje od takvog stanja. Prikupljanje podataka s oštećenih dijelova strojeva bi samo po sebi predstavljalo problem ako se ne radi o prikupljanju u kontroliranim, laboratorijskim uvjetima. Većina današnjih primjena strojnog učenja na realne probleme potpada u kategoriju polu-nadziranog učenja jer je često jako zahtjevno, skupo i vremenski neisplativo prikupljati i pravilno označavati sve vrste podataka.

U idućim poglavljima bit će detaljnije opisani neke od najkorištenijih platformi odnosno algoritama strojnog učenja.

### 3.1. Neuronske mreže

Jedan od najpoznatijih i danas najkorištenijih algoritama strojnog učenja su neuronske mreže. Ideja o računalnim neuronskim mrežama nije nova, već potiče čak od sredine 20. stoljeća kada su mnogi znanstvenici [17] potaknuti idejom o imitaciji biološke strukture mozga počeli razvijati takve algoritme.



**Slika 9 Slikoviti prikaz biološkog neurona (lijevo) i poveznica s umjetnim neuronom (desno). [13]**



Ipak, s vremenom dolazi do odmakta od ideje imitacije biološke neuronske mreže poglavito zbog nemogućnosti poznavanja točnog načina rada neurona, sinapsi i drugih dinamički kompliciranih dijelova neuronske mreže. Određene paralele se ipak mogu povući između biološkog i umjetnog neurona, barem kada je riječ o slikovitoj strukturi, kao što je prikazano na Slika 9. Osnovna funkcionalna jedinica neuronske mreže je **neuron**. Svaki neuron prima ulazne signale preko ulaznih kanala, biološki zvani dendriti, te procesira taj signal k izlazu preko izlaznog kanala aksona. Akson se potom grana i preko sinapsi spaja s drugim neuronima te time čini neuronsku mrežu. U strukturi umjetnog neurona, ulazi u neuron su izlazi iz prethodnih neurona  $x_0, x_1, \dots, x_j$ , koji s neuronom komuniciraju produktima  $w_0x_0, w_1x_1, \dots, w_jx_j$ , preko težina  $w_0, w_1, \dots, w_j$ . Ideja neuronske mreže se zasniva na pretpostavki da su težine adaptivne i mogu se naučiti na taj način da kontroliraju jakost i utjecaj jednog neurona na drugi. U prvom i najjednostavnijem modelu neurona, nazvanom perceptron [17] dendriti nose signal umnožen s težinama do neurona gdje se zbrajaju, a hoće li perceptron okinuti signal idućim neuronima, ovisi o tome je li zbroj ulaznih signala veći od određene granične vrijednosti (*eng. treshold*). Algebarski to se može prikazati kao:

$$output = \begin{cases} 0, & \text{if } \sum w_j x_j \leq threshold \\ 1, & \text{if } \sum w_j x_j > threshold \end{cases} \quad (2)$$

Ovakav model umjetnog neurona se pokušajem primjene u praksi pokazao nepouzdan i neučinkovit, poglavito zbog prevelike osjetljivosti na male promjene u ulazima neurona. Kako bi neuronska mreža efektivno učila potrebno je iterativnim postupkom biti moguće malo mijenjati težine kako bi se određenim algoritmom došlo do željenog rješenja. To se postiže uvođenjem aktivacijske funkcije kao nelinearne operacije izvedene na zbroju umnožaka ulaza i težina u neuronu.

### 3.1.1. Model neuronske mreže

Osnovna struktura neuronske mreže započinje definiranjem  $M$  linearnih kombinacija ulaza  $x_1, \dots, x_D$  i pripadajućih težina kojima se pribraja tzv. prag ili pomak (*eng. bias*) u obliku:

$$a_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (3)$$

gdje je  $j = 1, \dots, M$ , a indeks 1 referira se na činjenicu da je riječ o prvom sloju neuronske mreže. Navedene veličine  $a_j$  se nazivaju *aktivacije*, a one se transformiraju nelinearnim *aktivacijskim funkcijama*  $h(\cdot)$  kako bi dale:

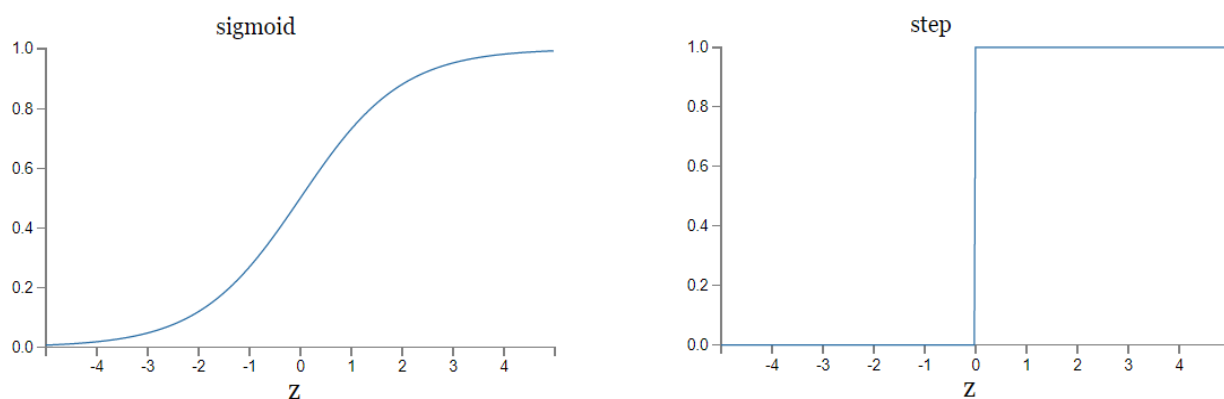
$$z_j = h(a_j) \quad (4)$$

Nakon prvog sloja, izlazne vrijednosti iz Jednadžbe (4) se ponovo linearno kombiniraju u idućem izlaznom sloju kako bi dale aktivacije izlaznih jedinica:

$$a_k = \sum_{i=1}^M w_{ki}^{(2)} z_j + w_{k0}^{(2)} \quad (5)$$

Naposljetku, izlazne aktivacije se transformiraju koristeći aktivacijske funkcije kako bi dobili set izlaznih vrijednosti mreže  $y_k$ . Izbor aktivacijske funkcije ovisi o samoj prirodi ulaznih podataka i pretpostavljenoj distribuciji ciljanih varijabli, a u početku se najviše koristila logistička sigmoidalna funkcija, koja se od primitivnog perceptrona razlikuje po vrijednostima koje izlaz aktivacijske funkcije može poprimiti, kao što je vidljivo na grafičkom prikazu na Slika 10. Sigmoidalna funkcija je definirana kao:

$$\sigma(a) = \frac{1}{1 + e^{-a}} \quad (6)$$



**Slika 10** Razlike između sigmoidalne funkcije (lijevo) i step funkcija korištene u prvom modelu perceptrona (desno)

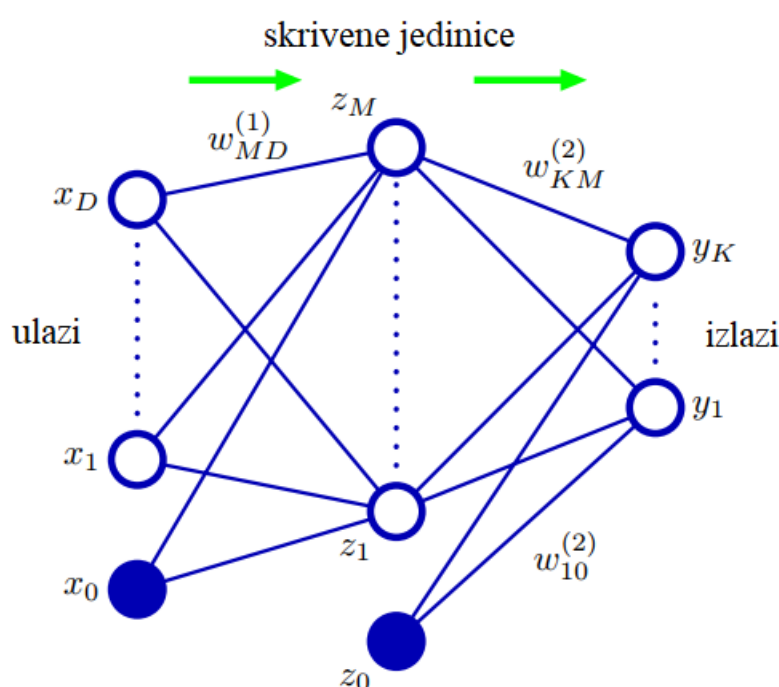
Ako prag  $w_{j0}^{(1)}$  iz prvog sloja i  $w_{k0}^{(2)}$  iz drugog sloja sažmemo unutar umnoška ulaza i težina tako što ćemo uz pragove vezat novu jedinicu  $x_0 = 1$ , onda Jednadžba (3) postaje:

$$a_j = \sum_{i=0}^D w_{ji}^{(1)} x_i \quad (7)$$

A slična se operacija može učiniti s produktom drugog sloja, te se ukupna izlazna funkcija neuronske mreže može zapisati na sljedeći način:

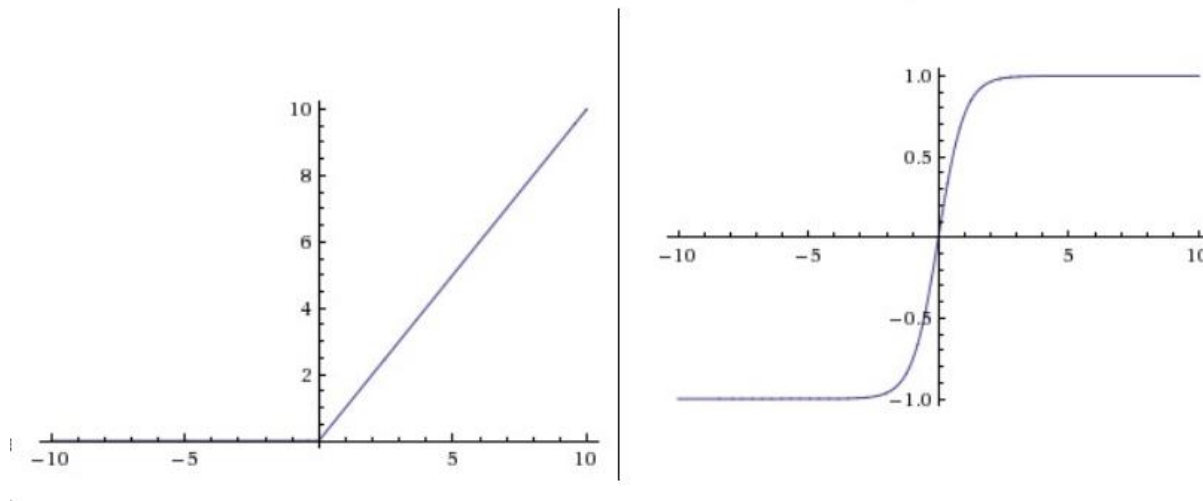
$$y_k(\mathbf{x}, \mathbf{w}) = \sigma \left( \sum_{i=0}^M w_{ki}^{(2)} z_j h \left( \sum_{i=0}^D w_{ji}^{(1)} x_i \right) \right) \quad (8)$$

Prethodna Jednadžba (8) predstavlja jednostavni model neuronske mreže koja je ustvari nelinearna funkcija od vektora ulaza  $\{x_i\}$  ka vektoru izlaza  $\{y_k\}$ , upravljana vektorom zajedničkim vektorom težina i pragova  $\mathbf{w}$ . Ovakva struktura mreže posjeduje ulazni sloj, jedan skriveni sloj i izlazni sloj, a grafički je prikazana na Slika 11. Ipak, ovakva mreža se naziva dvoslojna mreža, jer u samo dva sloja posjeduje vektore promjenjivih težina, skrivenom i izlaznom. Proces u kojem ulazni signal odnosno informacija propagira u samo jednom smjeru, od ulaza prema izlazu kao što je prikazano zelenim strelicama na Slika 11, naziva se proces unaprijedne propagacije signala te se onda ovakva neuronska mreža naziva unaprijedna ili aciklička neuronska mreža (eng. *Feed-forward Neural Network*).



Slika 11 Struktura neuronske mreže koja odgovara Jednadžbi (8) [16]

S obzirom da neuroni u prikazanoj strukturi na Slika 11 izgledaju isto kao i osnovni model perceptrona, ovakva neuronska mreža se u literaturi još i naziva **višeslojni perceptron** (eng. *MLP – Multilayer Perceptron*). Naravno, ključna razlika između perceptrona i neurona je u aktivacijskoj funkciji, gdje neuronska mreža koristi kontinuirane nelinearne funkcije u skrivenom sloju, dok perceptron koristi step funkciju - Slika 10. Nadalje, to znači da je funkcija neuronske mreže, jednadžba (8), diferencijabilna u odnosu na parametre mreže, a to svojstvo igra ključnu ulogu u procesu učenja mreže. Za sigmoidalnu funkciju se s vremenom pokazalo da posjeduje određena loša svojstva kao što je problem nestajućeg gradijenta, te izlaz koji nije centriran oko nule [18]. Zbog toga se češće koriste funkcije kao što su tanges hiperbolni ili zglobnica (eng. *Rectified Linear unit – ReLU*), prikazani na Slika 12.



**Slika 12** Prikaz zglobne aktivacijske funkcije ReLU (lijevo) i tangens hiperbolne funkcije (desno)

### 3.1.2. Učenje neuronske mreže

U svrhu učenja neuronske mreže prethodno je potrebno definirati funkciju gubitka. Definiranje funkcije gubitka ovisi o zadatku strojnog učenja. U neuronskim mrežama najčešće se teži minimiziranju funkcije gubitka, dok se u nekim drugim slučajevima može težiti maksimiziranju vjerojatnosti. Ako je zadatak učenja neuronske mreže regresija, onda se kao funkcija gubitka najčešće definira srednja kvadratna pogreška (eng. *Mean squared error – MSE*):

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 \quad (9)$$

gdje je  $y(x_n, \mathbf{w})$  algoritmom predviđena izlazna vrijednost, dok je  $t_n$  u ovom slučaju zadana, odnosno željena izlazna vrijednost.

Za zadatak binarne klasifikacije najčešće se koristi funkcija gubitka u obliku ukrštene entropije (eng. *Cross-Entropy*):

$$E(\mathbf{w}) = - \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \quad (10)$$

dok se u slučaju višerazredne klasifikacije između više međusobno različitih klasa  $K$  jednadžba (10) pretvara u drugi oblik:

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{k=1}^K \{t_{nk} \ln y_k(x_n, \mathbf{w})\} \quad (11)$$

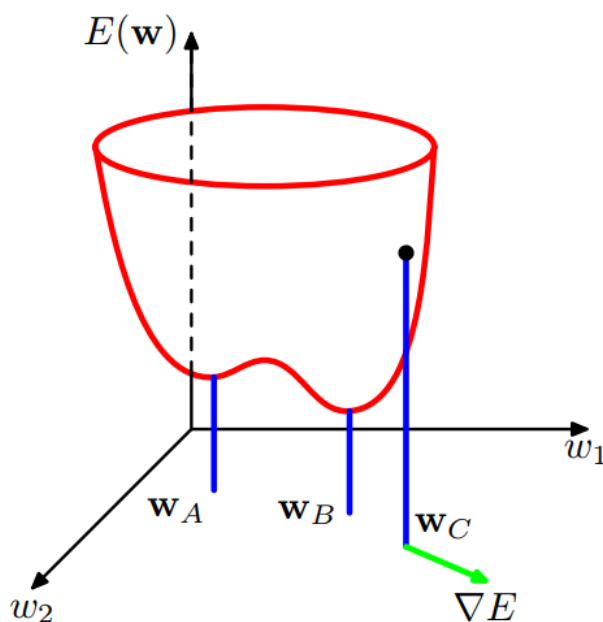
Nakon definiranja funkcije cilja, potrebno je da mreža pronađe vektor težina  $\mathbf{w}$  koji minimizira izabranu funkciju gubitka  $E(\mathbf{w})$ . Korisno je vizualizirati funkciju gubitka kao prostornu veličinu koja ovisi o težinama, kao što je slikovito prikazano na Slika 13. Promjenama vektora

težina želimo utjecati na funkciju cilja tako što ćemo pronaći minimum te funkcije. Ako vektor težina mijenjamo s malim iznosima:  $\mathbf{w} = \mathbf{w} + \delta\mathbf{w}$ , onda slijedi da je promjena funkcije cilja:

$$\delta E \cong \delta\mathbf{w}^T \nabla E(\mathbf{w}) \quad (12)$$

gdje je  $\nabla E(\mathbf{w})$  gradijentni vektor funkcije gubitka i pokazuje u smjeru najveće stope rasta funkcije gubitka. Najmanja vrijednosti funkcije gubitka nastupit će na mjestu gdje će gradijent funkcije biti jednak:

$$\nabla E(\mathbf{w}) = 0 \quad (13)$$



**Slika 13** Grafički prikaz odnose funkcije cilja i vektora težina s dvije vrijednosti. [16]

Mjesta na kojima gradijent nestaje nazivaju se stacionarnim točkama. Prikaz na Slika 13 je prilično jednostavan, jer funkcija cilja je uobičajeno izrazito nelinearna i kompleksna funkcija koja ovisi o mnogo težina i pragova. Praktično je nemoguće analitički riješiti jednadžbu (13), tako da se problem minimiziranja funkcije gubitka obično rješava iterativnim numeričkim postupcima gdje se izabire inicijalna vrijednost vektora težina koja se potom mijenja u koracima

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^\tau + \Delta\mathbf{w}^\tau \quad (14)$$

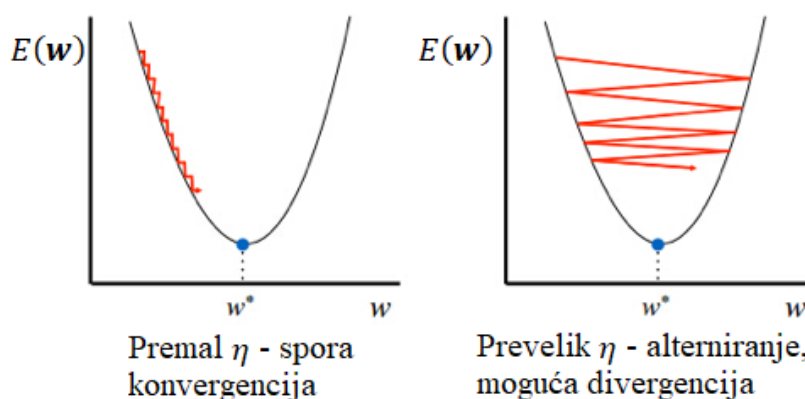
gdje  $\tau$  označava trenutni iterativni korak. Parametre odnosno težine o kojima funkcije greške ovisi mijenjamo u smjeru suprotnom od smjera vektora gradijenta funkcije gubitka (minus u jednadžbi (15)). Mnogi algoritmi optimizacije funkcije cilja koriste informaciju o gradijentu funkcije, a najpoznatiji algoritam je gradijenti spust (*eng. Gradient descent*) ili njegova varijacija stohastički gradijenti spust gdje se umjesto računanja funkcije gubitka na cijelom

skupu podataka, odabiru manji, zasebni skupovi podataka (*eng. mini batch*), na kojima se provodi učenje. Kada se jednom završi učenje na svim manjim skupovima podataka, takav ciklus naziva se *epoha*. Jednadžba stohastičkog gradijentnog spusta:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^\tau - \eta \nabla E_n(\mathbf{w}^\tau) \quad (15)$$

gdje je faktor  $0 < \eta < 1$  nazvan stopa učenja. Stopa učenja jedan je od glavnih hiperparametara neuronske mreže kojeg treba samostalno izabrati kako bi mreža postigla dobru konvergenciju prema najboljem vektoru težina. Tri su najčešća slučaja izbora stope učenja [19]:

- Premala stopa učenja rezultira monotonom, ali jako sporom konvergencijom.
- Stopa učenja je velika, ali manja od maksimalne. Rezultat je alternirajuće približavanje idealnoj vrijednosti  $\mathbf{w}$ .
- Stopa učenja je veća od maksimalne i rezultat učenja je divergencija od dobrog rješenja. Ovo je ujedno i najnepovoljniji slučaj kojim se udaljavamo od minimalne vrijednosti funkcije gubitka.



**Slika 14** Različiti odabiri stope učenja utječu na konvergenciju rješenja

Kao što je prikazano na Slika 13, funkcija greške u ovisnosti o težinama može biti geometrijski prikazana kao hiperpovršina nad težinama koja sadrži mnogo lokalnih minimuma ( $w_A$ ) u kojima gradijent nestaje ili poprima jako malu vrijednost, te se algoritam gradijentnog spusta zaustavlja i prije globalnog minimuma. Iako se u praksi pokazalo da pronalazak globalnog minimuma nije nužan uvjet efikasnog rada neuronske mreže, postoji tehnika primjene dodatnog hiperparametra mreže, momenta inercije učenja, kojim se u nekim slučajevima izbjegava upadanje u lokalne minimume.

### 3.1.3. Algoritam povratnog prostiranja pogreške

Kako bi se kompletirao algoritam neuronske mreže potrebno je izračunati gradijent funkcije pogreške. Algoritam poznat kao povratno prostiranje pogreške (*eng. Error backpropagation / Backprop*) osmišljen još u 70-tim godinama prošlog stoljeća, uveden je u neuronske mreže prvi put 1986. god [20] i otada se zadržao kao najčešći oblik računanja gradijenta funkcije gubitka. U postupku učenja mreže prvotno je potrebno odrediti derivacije funkcije greške u odnosu na težine i pragove, a pomoću algoritma prosljeđivanja greške unazad taj se postupak može lagano implementirati u računalne programe. U drugoj fazi se te derivacije koriste za prilagođavanje težina.

U općenitoj unaprijednoj mreži sa Slika 11, svaka jedinica računa težinsku sumu ulaza:

$$a_j = \sum_i w_{ji} z_i \quad (16)$$

gdje je  $z_i$  ulaz ili prethodna aktivacijska jedinica koja šalje vezu jedini  $j$ , a  $w_{ji}$  je pripadajuća težina. Suma iz prethodne jednadžbe prolazi kroz nelinearnu transformaciju:

$$z_j = h(a_j) \quad (17)$$

Parcijalna derivacija funkcije gubitka po vektoru težina se može zapisati pomoću lančanog pravila:

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \quad (18)$$

gdje je  $a_j$  vektorski umnožak težina i ulaza u jedan neuron, definiran u jednadžbi (7) za prvi sloj ili u jednadžbi (8) za izlazni sloj. Derivacija funkcije greške po tom vektorskom umnošku može se zamijeniti s oznakom koja se naziva *greška* u  $j$ -tom neuronu:

$$\delta_j = \frac{\partial E_n}{\partial a_j} \quad (19)$$

Koristeći jednadžbu (16), dobijamo:

$$\frac{\partial a_j}{\partial w_{ji}} = z_i \quad (20)$$

te uvrštavajući jednadžbe (19) i (20) u (18) dobiva se izraz za derivaciju funkcije greške:

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \quad (21)$$

Iz prethodne jednadžbe vidimo da treba odrediti grešku  $\delta_j$ . Greška u izlaznom sloju se može izraziti kao razlika između predviđene i željene veličine:

$$\delta_k = y_k - t_k \quad (22)$$

Koristeći jednadžbu (19) i lančano pravilo deriviranja, dolazimo do jednadžbe prosljeđivanja pogreške unazad koja glasi:

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \quad (23)$$

S obzirom da grešku u posljednjem sloju definiranu jednadžbom (22) uvijek možemo odrediti, greške u jedinicama bilo kojeg skrivenog sloja možemo odrediti tako što propagiramo grešku iz najvišeg sloja unazad preko jednadžbe (23), što je i smisao *backprop* algoritma. Koraci algoritma su sljedeći:

1. Narinuti vektor ulaznih veličina  $x_n$  na neuronsku mrežu te propagirati informaciju unaprijed preko jednadžbi (16) i (17) kako bi pronašli aktivacije skrivenih i izlaznih jedinica
2. Odrediti grešku u izlaznom sloju  $\delta_k$  pomoću jednadžbe (22)
3. Prostirati greške  $\delta$  unazad koristeći jednadžbu (23) kako bi se pronašle greške u skrivenim jedinicama.
4. Pomoću jednadžbe (21) izračunati tražene derivacije

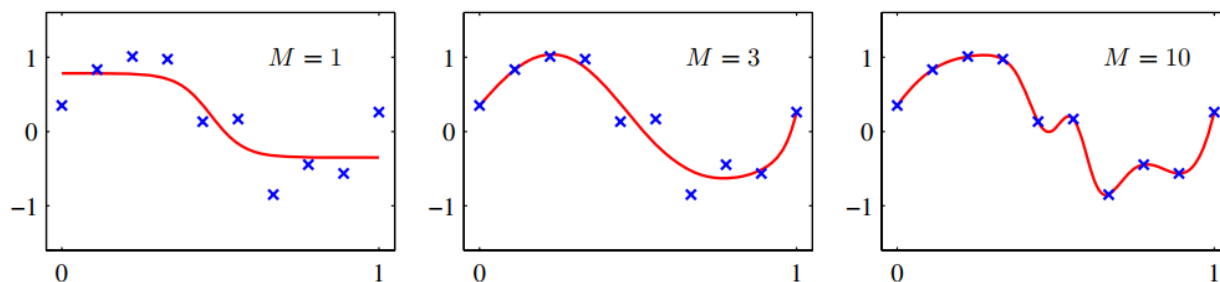
#### 3.1.4. Regularizacija neuronske mreže

Uspješnost rada algoritma strojnog učenja na novim, za algoritam neviđenih podataka smatra se općenito centralnim ciljem strojnog učenja. Sposobnost obavljanja danog zadatka na neviđenim ulazima naziva se *generalizacija*. Tijekom treniranja modela strojnog učenja, računa se greška treniranja koja se pokušava smanjiti. Ono što razdvaja strojno učenje od optimizacije je činjenica da model strojnog učenja mora smanjiti i testnu odnosno generalizacijsku pogrešku. Uspješnost algoritma strojnog učenja, pa tako i neuronske mreže ocjenjuje njegova mogućnost da:

1. Učini grešku na trening setu podataka malom
2. Smanji razliku između greške trening i testnih podataka

Osnovno svojstvo algoritama strojnog učenja je *kapacitet*. Kapacitet algoritma opisuje njegovu mogućnost prilagođavanja raznim podacima, te je najčešće ovisan o dimenzionalnosti ulaznih podataka. Pronalazak prave vrijednosti kapaciteta neuronske mreže je čest problem pri dizajniranju hiperparametara. Modeli malog kapaciteta su skloni podnaučenosti, dok su modeli velikog kapaciteta skloni prenaučeniosti (*eng. underfitting, overfitting*). Oba slučaja loše utiču na sposobnost generalizacije algoritma, kao što je prikazano na Slika 15.



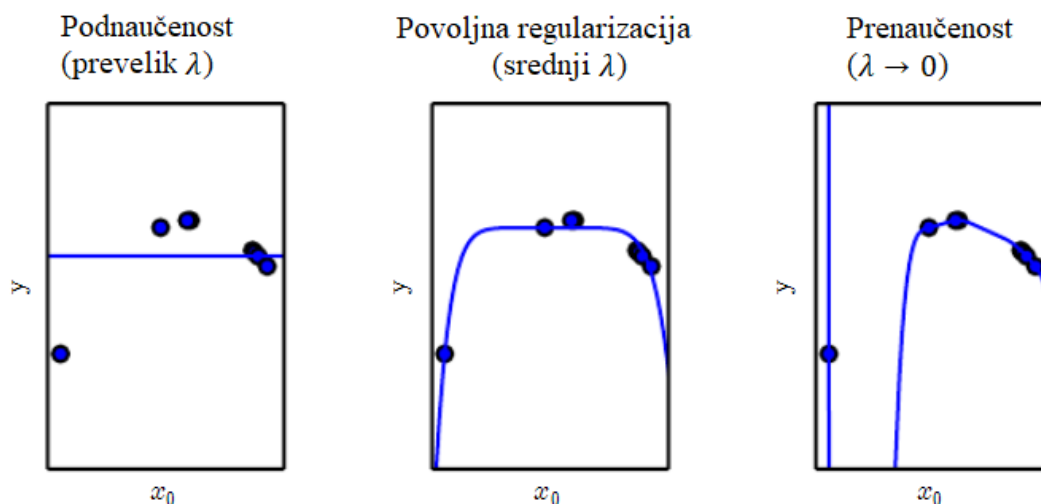


**Slika 15** Primjer učenja dvoslojne neuronske mreže trenirane na 10 trening podataka iz sinus funkcije.  $M$  označava broj neurona u skrivenom sloju. Podnaučenost lijevo, generalizacija u sredini, prenaučenosť desno.

Ipak, promjena dimenzionalnosti ulaznih podataka ili jedinica u skrivenom sloju nije dobar način rješavanja problema loše generalizacije jer time gubimo željene podatke i smanjuje se efikasnost mreže. Umjesto toga, kapacitet neuronske mreže povećava se regularizacijom algoritma. Regularizacija se najčešće primjenjuje na funkciju gubitka, tako što se uvodi tzv. kažnjavanje visokih vrijednosti težina (eng. *weight decay*):

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (24)$$

Uvođenjem regularizatora  $\mathbf{w}^T \mathbf{w}$  i faktora regularizacije  $\lambda$  penaliziramo funkcije s visokim težinama te potičemo algoritam da preferira funkcije s manjim težinama.

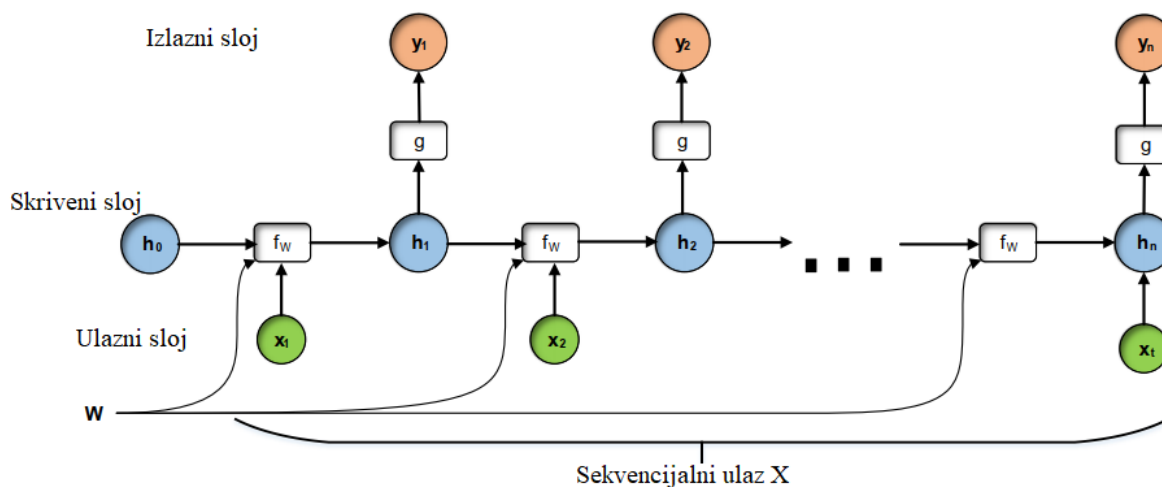


**Slika 16** Utjecaj faktora regularizacije  $\lambda$  na kapacitet modela

### 3.2. Povratne neuronske mreže

Povratne neuronske mreže (eng. *Recurrent Neural Networks – RNN*), za razliku od unaprijednih neuronskih mreža, ne uzimaju za ulaz samo informacije koje dolaze određenim slijedom, nego i informacije koje su obradili prethodno nekoliko koraka unatrag. Kombinirajući te informacije, povratne mreže donose odluke na osnovu više prethodno viđenih podataka te su takvom

topologijom povezanosti neurona prikladne za učenje značajki iz sekvencijalnih podataka kao što je tekst (riječi, rečenice) ili podataka vremenske serije.



Slika 17 Struktura povratne neuronske mreže [22]

Za povratne neuronske mreže može se reći da posjeduju memoriju [23] koja pomaže pri lakšoj obradi i prepoznavanju sekvencijalnih podataka, s obzirom da u takvim sljedovima nisu povezani samo susjedni podaci, nego i oni u „daljoj prošlosti“. Obradena informacija se čuva u skrivenim jedinicama povratne neuronske mreže te utječe na obradu novo pridošlih ulaza. Pomoću toga pronalazi korelacije između ulaza razdvojenih više trenutaka koje se nazivaju „dugoročne korelacije“, jer događaj nakon nekog vremena u sekvencijalnom podatku ovisi o događaju koji se dogodio dublje u prošlosti. Matematički se izlazna informacija neurona u povratnoj mreži može izraziti:

$$\mathbf{h}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1}) \quad (25)$$

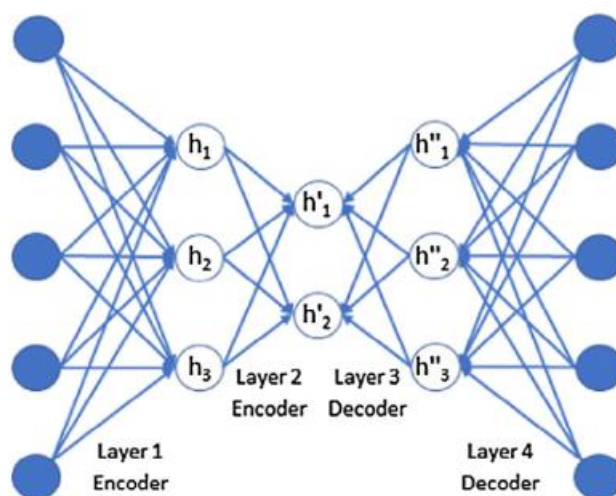
Skriveno stanje u vremenskom trenutku  $t$  je veličina  $\mathbf{h}_t$  koja ovisi o ulazima u istom vremenu  $t$  i pripadajućem vektoru težina  $\mathbf{W}$ , ali i o skrivenom stanju u vremenskom trenutku  $t-1$ , i pripadajućoj matrici tranzicije iz jednog skrivenog u drugo skriveno stanje [isto ko i prošlo]. Matrice težina i tranzicijska matrica služe kao parametri koji određuju koliku važnost treba pridati sadašnjem ili prošlom stanju, a greška koju uzrokuju će se vratiti algoritmom prilagođenim algoritmom povratnog prostiranja pogreške (poglavlje 3.1.3) zvan povratno prostiranje pogreške u vremenu (*eng. BackProp Through Time*) i iskoristiti za prilagođavanje njihovih vrijednosti dok greška ne padne na minimalnu vrijednost.

Korištenjem takvog algoritma otkriven je problem nestajućeg i eksplodirajućeg gradijenta koji je učenje povratne neuronske mreže činio jako teškim ili nemogućim, pogotovo na dugačkim sekvencijalnim podacima [23]. Kako bi se riješio taj problem, sredinom 90-tih godina prošlog

stoljeća izmišljene su, kao varijacija, povratne neuronske mreže s *jedinicama za dugotrajno kratkoročno pamćenje* (eng. *Long Short-Term Memory units - LSTMs*). LSTM jedinice pomažu pri održavanju konstantnije vrijednosti pogreške tijekom učenja te tako daju mogućnost povratnoj mreži da uči u mnogo više koraka bez da gradijent nestane ili eksplodira. LSTM jedinica se može shvatiti kao ćelija s tri različita stanja ili vrata, a to su *ulaz, izlaz i brisanje*. Svako od stanja ili vrata ima vlastiti vektor naučenih težina na osnovu kojeg odlučuje treba li ulazni signal dopustiti ulaz u iterativni proces, izlaz iz procesa ili ga treba izbrisati iz procesa. Korištenje LSTM jedinica rješava problem nestajućeg gradijenta, ali očito unosi mnogo više parametara potrebnih za naučiti tijekom procesa treniranja mreže.

### 3.3. Autoenkoder

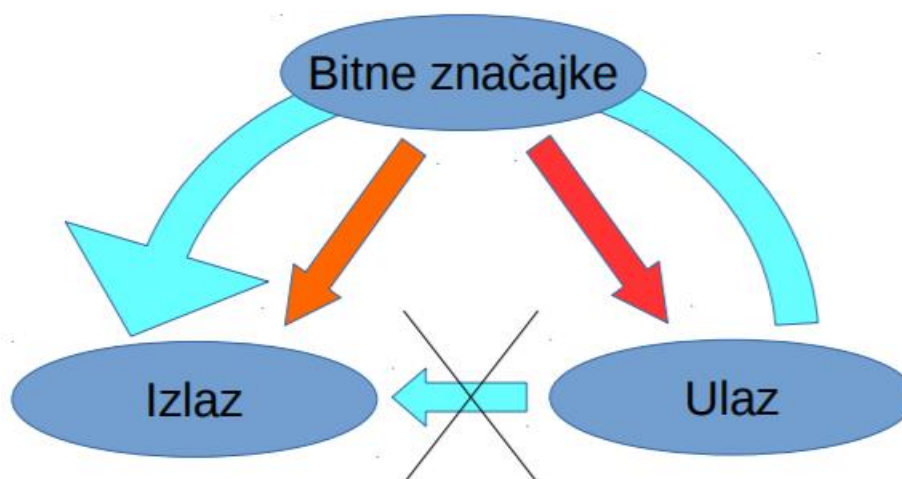
Autoenkoder je algoritam nenadziranog učenja pomoću kojeg je moguće izvlačiti značajke iz ulaznog signala bez dodatnih informacija o klasi ili oznaci tog signala [24]. Obični autoenkoder sastoji se iz dva dijela, enkodera i dekodera, odnosno ulaznog, skrivenog/skrivenih i izlaznog sloja.



Slika 18 Struktura autoenkodera [24]

Enkoder vrši kompresiju visoko-dimenzionalnih podataka preslikavajući ulaz na skriveni sloj, a time skriveni sloj uči informacije o ulaznim podacima bez pomoći odnosno labelizacije podataka. Naučene informacije spremaju se kao težine skrivenog sloja. Dekoder vrši rekonstrukciju aproksimirajući ulazni signal iz skrivenog sloja, a zadatak cijelog autoenkodera je naučiti set težina koji će najtočnije aproksimirati zadani ulaz. Što je ulazni signal kompleksniji i s više dimenzija, to je potrebno više skrivenih slojeva unutar autoenkodera kako bi se pravilno rekonstruirale mnoge nelinearnosti signala. Sami izlaz autoenkodera nije toliko zanimljiv jer je to samo vjerodostojna kopija ulaza, ali značajke unutar skrivenog sloja koje

karakteriziraju ulaz su ono što je bitno kod rezultata autoenkodera. Tako se autoenkodera mogu koristiti kao nenadzirani algoritam prije učenja nekog drugog nadziranog algoritma, ako ulazni podaci nisu potpuno označeni.



Slika 19 Dijagram rekonstrukcija signala unutar autoenkodera

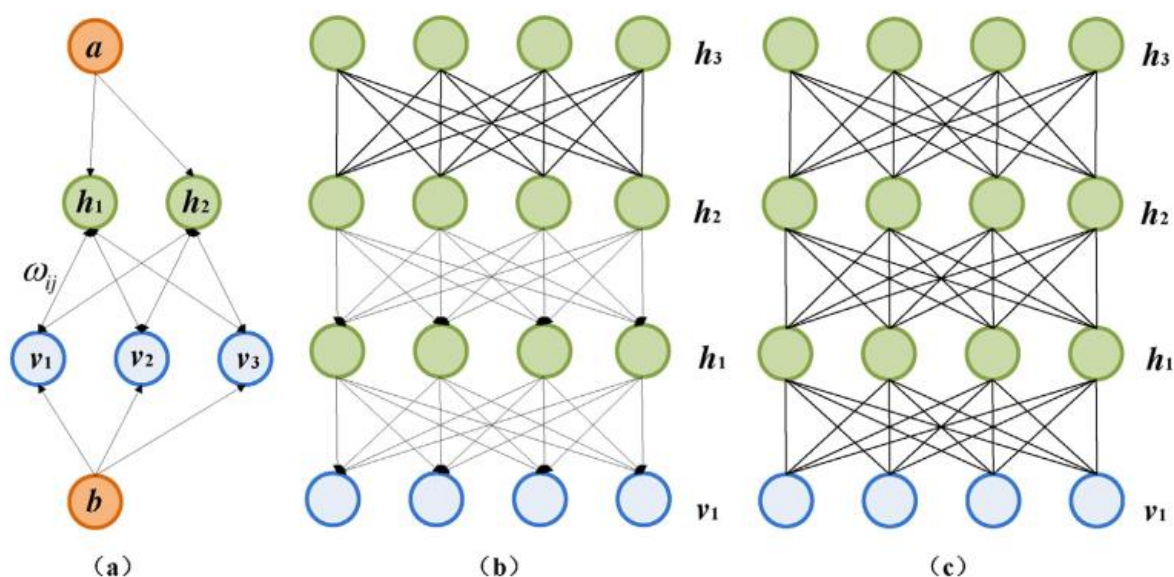
### 3.4. Ograničeni Boltzmannovi strojevi

Algoritam ograničenih Boltzmannovih strojeva (*eng. Restricted Boltzmann Machines – RBM*) je vrsta „plitke“ neuronske mreže ili vrsta autoenkodera pogodna za redukciju dimenzionalnosti podataka, regresiju, klasifikaciju te učenje značajki. Ograničeni Boltzmannovi strojevi mogu se shvatiti kao dvoslojna neuronska mreža koja se sastoji od jednog vidljivog i jednog skrivenog sloja. Naziv „ograničeni“ potiče od činjenice da za razliku od običnih Boltzmannovih strojeva (BM), ukinute su bilo kakve veze neurona unutar istog sloja. RBM algoritam spada u skupinu generativnih modela zbog toga što mogu naučiti raspodjelu vjerojatnosti ulaznih podataka te potom generirati nove uzorke koji odgovaraju skupu trening podataka. Kod problema nepotpunih ulaznih podataka, RBM algoritam može na osnovu predviđanja raspodjele vjerojatnosti nadomjestiti izgubljene ili nedostajuće ulazne podatke.

Težine i pomaci između dva sloja uče se iteracijama kako bi izlaz vidljivog sloja bio što bolja aproksimacija ulaznog signala. Parametri u skrivenom sloju mogu biti promatrani kao značajke koje karakteriziraju ulazni signal bez prethodnog znanja o prirodi signala, a potom se na RBM algoritam mogu postaviti algoritmi nadziranog učenja kao što je logistička regresija ili neuronske mreže koje mogu izvršiti klasifikaciju ili regresiju.

Kao samostalni algoritam, RBM se mogu slagati jedan na drugi i tako načiniti algoritme zvane duboke probabilističke mreže (*eng. Deep Belief Networks – DBN*) ili duboki Boltzmannovi

strojevi (eng. *Deep Boltzmann Machine – DBM*). Jedina razlika između ta dva algoritma je u njihovom načinu spajanja slojeva. Duboke probabilističke mreže su djelomično usmjerene mreže gdje su sve veze osim onih između dva najviša sloja usmjerene prema dolje, dok su duboki Boltzmannovi strojevi potpuno usmjereni algoritmi. Usmjerenost veze podrazumijeva da neuron od kojeg veza kreće utječe na neuron u kojeg ulazi, ali ne i obrnuto. Prema tome, algoritam duboke probabilističke mreže može biti efikasniji i lakši za naučiti od dubokog Boltzmannovog stroja. Detaljniji opis algoritama može se naći u [25]. Strukture opisanih algoritama prikazane su grafički na Slika 20.



Slika 20 Arhitektura a) Ograničenog Boltzmannovog stroja, b) Duboke probabilističke mreže, c) Dubokog Boltzmannovog stroja [22]

### 3.5. Duboke konvolucijske neuronske mreže

Neuronske mreže i varijacije s višeslojnim perceptronima su zasigurno najsnažniji algoritam strojnog učenja zbog činjenice da su neuronske mreže univerzalni aproksimatori [26] [27]. Značenje univerzalnog aproksimatora je sljedeće: *Bez obzira koliko je funkcija složena, postoji neuronska mreža s određenim brojem slojeva i neurona čiji izlaz može biti dovoljno dobra aproksimacija te funkcije.* S povećanjem opsega problema, povećava se potreban broj računskih jedinica neurona unutar mreže, a time se posljedično povećava i težina i vrijeme izvođenja računalnih operacija unutar procesorske jedinice. Razvojem mikroprocesorske industrije došlo je do mogućnosti za primjenu algoritama dubokog strojnog učenja, ponajviše dubokih neuronskih mreža odnosno jedne od inačica, konvolucijskih neuronskih mreža.

Konvolucijske neuronske mreže (eng. *Convolutional Neural Networks – CNN*) su posebna vrsta neuronskih mreža koja je svojom internom strukturom prilagođena za relativno brzo i efikasno

rješavanje problema visoko-dimenzionalnih ulaza odnosno ulaza koji posjeduju mnoštvo značajki. Svoju glavnu primjenu CNN su stekle u obradi, klasifikaciji i prepoznavanju objekata na slikama. Prvotnu implementaciju napravio je 1998. god. Yann LeCun [28] na primjeru automatskog prepoznavanja ručno pisanih znamenki s fotografija. Konvolucijske neuronske mreže su se od početka korištenja u praksi pokazale kao revolucionarno rješenje za obradu najzahtjevnijih zadataka. Osim što su dale najbolje rezultate u prepoznavanju slika [29] i govora [30], duboke neuronske mreže su vodeće u primjeni u medicini [31], genetici [32] i raznim prirodnima znanostima. Dva glavna razloga zašto su konvolucijske neuronske mreže omogućile najnovija dostignuća u primjeni umjetne inteligencije su:

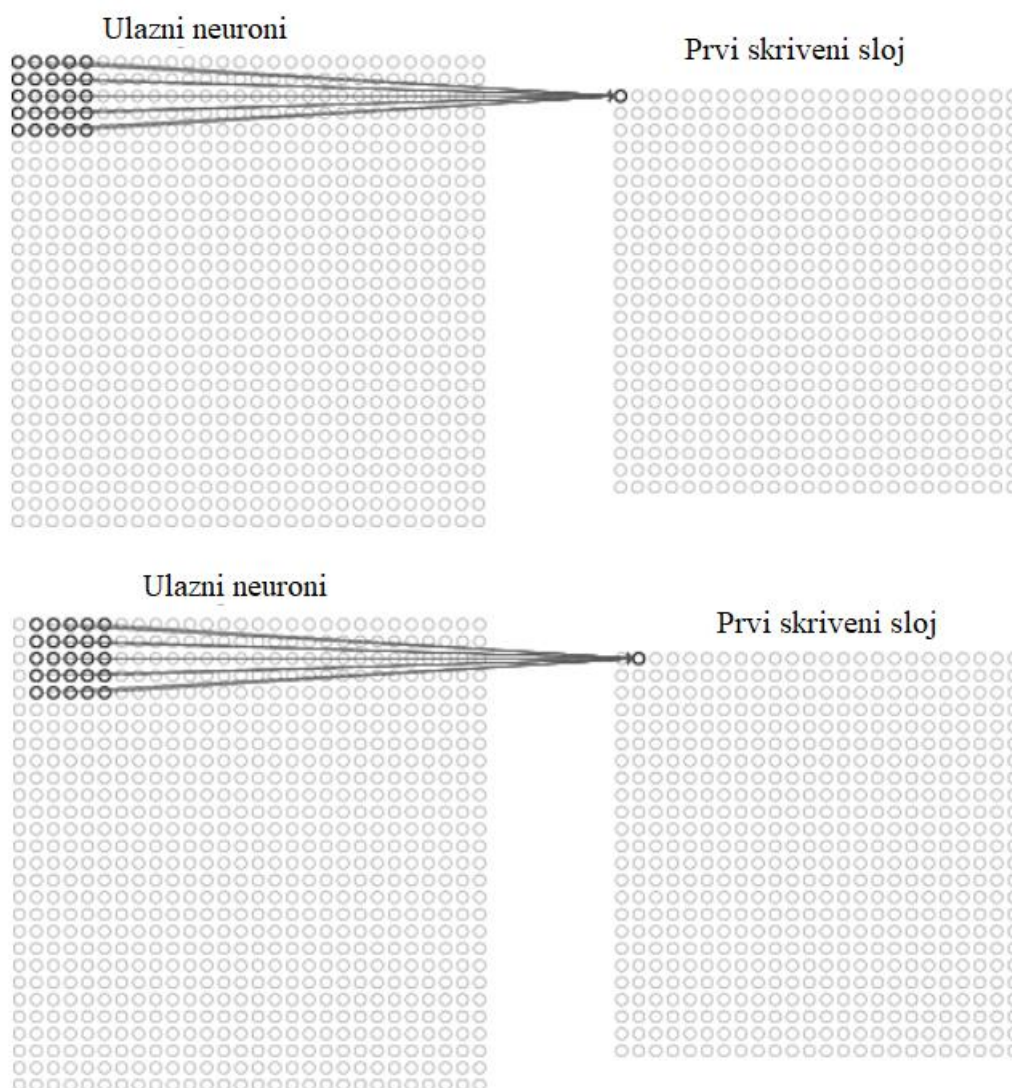
- Invarijantnost na određene transformacije i promjene ulaznog signala. Ova karakteristika je ključna u obradi fotografija ili vremenskog signala, zbog toga što takve datoteke mogu biti značajno izmijenjene (rotacija objekta na slici, šum u vremenskom signalu), a njihovo značenje ostaje nepromijenjeno. Također, neke vrste značajki se ponavljaju na različitim mjestima unutar datoteke, te je potrebno obratiti pozornost na njihovo značenje i pravilno ih prepoznati u bilo kojem dijelu datoteke.
- Mogućnost automatskog izvlačenja značajki iz izvornih datoteka. S ovom karakteristikom duboko učenje omogućuje da ulazni signal u mrežu bude izvorna datoteka (fotografija iz kamere) ili sirovi signal (signal sa senzora zvuka, vibracija...). Nije potrebno mukotrpno ručno izvlačenje značajki koje oduzima mnogo vremena te zahtjeva strogo i detaljno poznavanje domene koja se obrađuje. Duboko učenje omogućuje reprezentativno učenje, tj. učenje složenih reprezentacija iz izvornog, sirovog signala.

Konvolucijske neuronske mreže strukturirane su tako da najbolje reagiraju na ulaz u obliku višedimenzionalnih polja. Na primjer, fotografija u boji sadrži tri (crvena, zelena i plava) 2D polja koja sadrže intenzitete piksela, signali i vremenske serije su u obliku 1D polja, dok su video snimke i volumetrijske slike primjeri 3D polja. Postoje četiri ključne karakteristike konvolucijskih neuronskih mreža zbog kojih tako efikasno iskorištavaju svojstva prirodnih signala:

- Lokalna receptivna polja / Lokalne interakcije
- Dijeljenje parametara odnosno težina
- Sažimanje
- Slaganje više slojeva



Ako ulaz u mrežu promatramo kao vektor, matricu ili polje određenih vrijednosti, onda je veličina tog polja ujedno i veličina prvog ulaznog sloja konvolucijske neuronske mreže. Kao i kod običnih neuronskih mreža, ulazni sloj se povezuje s neuronima prvog skrivenog sloja, ali umjesto spajanja svih neurona kao na Slika 11, svaki neuron skrivenog sloja se spaja samo s manjom grupom neurona iz ulaznog sloja. Regija neurona s kojom se jedinica skrivenog sloja povezuje naziva se *lokalna receptivna regija* ili *polje* [33]. Polje se određenim korakom pomiče po ulaznom sloju te je svaki neuron skrivenog sloja produkt zasebnog ulaznog polja. Pomicanje polja po ulaznom sloju može se vizualizirati na primjeru 2D ulaznog polja na Slika 21.



**Slika 21** Povezivanje lokalnih receptivnih polja i skrivenih neurona [33]

Na prethodnoj slici, odmah se može uočiti da se uvođenjem lokalnih receptivnih polja, veličina sloja smanjuje kako signal propagira prema naprijed. Također, na Slika 21, lokalno polje se pomiče korakom od jednog neurona, ali to ne mora biti nužan slučaj. Veličina polja i korak pomicanja može biti proizvoljno odabran hiperparametar CNN-a.

Nakon učenja, vrijednosti težina i pragova koji povezuju polja ulaznih neurona i skrivene neurone imati će iste vrijednosti. To je druga bitna karakteristika konvolucijskih neuronskih mreža, zvana dijeljenje parametara (*eng. weight sharing*). Algebarski, vrijednost skrivenih  $j, k$  neurona se izražava formulom:

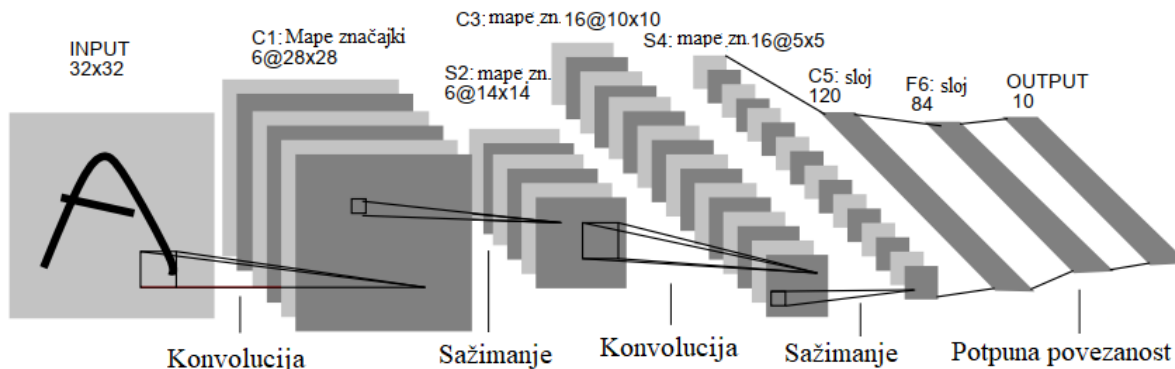
$$z_{j,k} = h \left( b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m} \right) \quad (26)$$

gdje je  $h(\cdot)$  bilo koja nelinearna aktivacijska funkcija,  $b$  zajednička vrijednost praga,  $w_{l,m}$  je receptivno polje težina veličine  $5 \times 5$ . Činjenica da čitav skriveni sloj dijeli iste vrijednosti naučenih parametara znači da čitav skriveni sloj prepoznaje i uči istu značajku, na svim mogućim lokacijama izvorne datoteke. Operacija iz jednadžbe (26) se matematički naziva konvolucija, pa otuda i naziv za konvolucijske neuronske mreže.

Pomoću navedenih karakteristika, konvolucijske neuronske mreže mogu automatski učiti razne vrste značajki, od najjednostavnijih i primitivnih do složenih i kompleksnih. Zbog toga se mapa neurona skrivenog sloja naziva i mapa značajki i nije ograničena na samo jednu značajku. U istom skrivenom sloju, moguće je naučiti više mapa značajki, od kojih će svaka imati zaseban set naučenih parametara koji će aktivirati određenu značajku prilikom njenog pronalaska na ulaznom signalu. Set naučenih parametara (težine i pragovi) nazivaju se *filteri* ili *jezgre* (*eng. kernel*).

Ključna razlika u strukturi konvolucijskih neuronskih mreža je moguća trodimenzionalnost unutar slojeva. Ulazne datoteke mogu biti sačinjene od više kanala, a skriveni slojevi mogu sadržavati više zasebnih mapa značajki, što omogućuje učenje različitih značajki. Efektivnost konvolucijskih neuronskih mreža proizlazi iz mogućnosti slaganja različitih vrsta slojeva, čime se postiže dubina mreže. Unutar mreže koriste se konvolucijski, aktivacijski, sažimajući slojevi, a na kraju mreža najčešće sadrži jedan ili više potpuno povezanih slojeva, te softmax funkciju za klasifikaciju. Trodimenzionalnost strukture je očita na Slika 22, gdje su prikazani slojevi neuronske mreže. Vidljivo je slaganje slojeva i postavljanje više mapa značajki u jednom sloju, kao npr. 6 mapa značajki u prvom sloju, a 16 mapa značajki u trećem sloju. Na kraju mreža posjeduje potpuno povezane slojeve izlaznih jedinica.





Slika 22 Primjer strukture konvolucijske neuronske mreže LeNet-5 iz [28]

**Konvolucijski sloj** je prvi skriveni sloj u strukturi konvolucijske neuronske mreže, a njegov način rada je opisan u prethodnom dijelu. Jednadžba (26) se može zapisati i općenitijim izrazom:

$$a^1 = h(b + w * a^0) \tag{27}$$

U konvolucijskim slojevima, izabiru se hiperparametri mreže kao što su veličina filtra i broj filtera odnosno broj mapa značajki. Korištenjem lokalnih receptorskih polja i dijeljenja parametara, postiže se smanjenje broja parametara koje mreža mora naučiti, te smanjenje broja računskih operacija u odnosu na potpuno povezanu, tradicionalnu neuronsku mrežu.

Konvolucijski sloj	
ULAZ	$W_1 \times H_1 \times D_1$
HIPERPARAMETRI	Broj filtera, $K$
	Veličina filtera, $F$
	Korak filtera, $S$
	Nadopuna, $P$
IZLAZ	$W_2 \times H_2 \times D_2$
VELIČINA	$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
	$H_2 = W_2$ , za 2D ulaz; $H_2 = 1$ , za 1D ulaz
	$D_2 = K$

Tablica 2 Dimenzije konvolucijskog sloja i pripadajućih hiper-parametara

Poslije sloja konvolucije ulaza preko filtera, dobivene mape značajki potrebno je kao i u potpuno povezanim neuronskim mrežama provesti kroz nelinearne aktivacijske funkcije kako bi se modelirale nelinearnosti koje su nužna pojava u značajkama. Najčešće aktivacijske funkcije spomenute su u poglavlju 3.1.1, a najpovoljnije rezultate u konvolucijskim neuronskim mrežama daje zglobna aktivacijska funkcija ili ReLU, prikazana na Slika 12.

Poslije konvolucije i nelinearne aktivacije, uvodi se **sloj sažimanja** (eng. *pooling, subsampling layer*). Sažimanje je jednostavna operacija smanjivanja dimenzionalnosti prethodnog sloja, tako što se skupine susjednih piksela zamjenjuju prosječnom vrijednosti neurona skupine (prosječno sažimanje – eng. *average pooling*) ili maksimalnom vrijednosti (maksimalno sažimanje – eng. *max pooling*). Intuicija koja stoji iza sažimajućeg sloja je ta da u konvolucijskoj neuronskoj mreži nije toliko bitna egzaktna lokacija određene značajke, nego samo njena relativna pozicija u odnosu na druge značajke. Sažimanjem se povećava invarijantnost mreže, sprječava se prenaučenosť uslijed velikih dimenzija i značajno se smanjiva potreban broj adaptivnih parametara (težina i pragova).

Sažimajući sloj	
ULAZ	$W_2 \times H_2 \times D_2$
HIPERPARAMETRI	Veličina filtera, $F$
	Korak filtera, $S$
IZLAZ	$W_3 \times H_3 \times D_3$
VELIČINA	$W_3 = \frac{(W_2 - F)}{S} + 1$
	$H_3 = W_3$ , za 2D; $H_3 = 1$ , za 1D
	$D_2 = D_3$

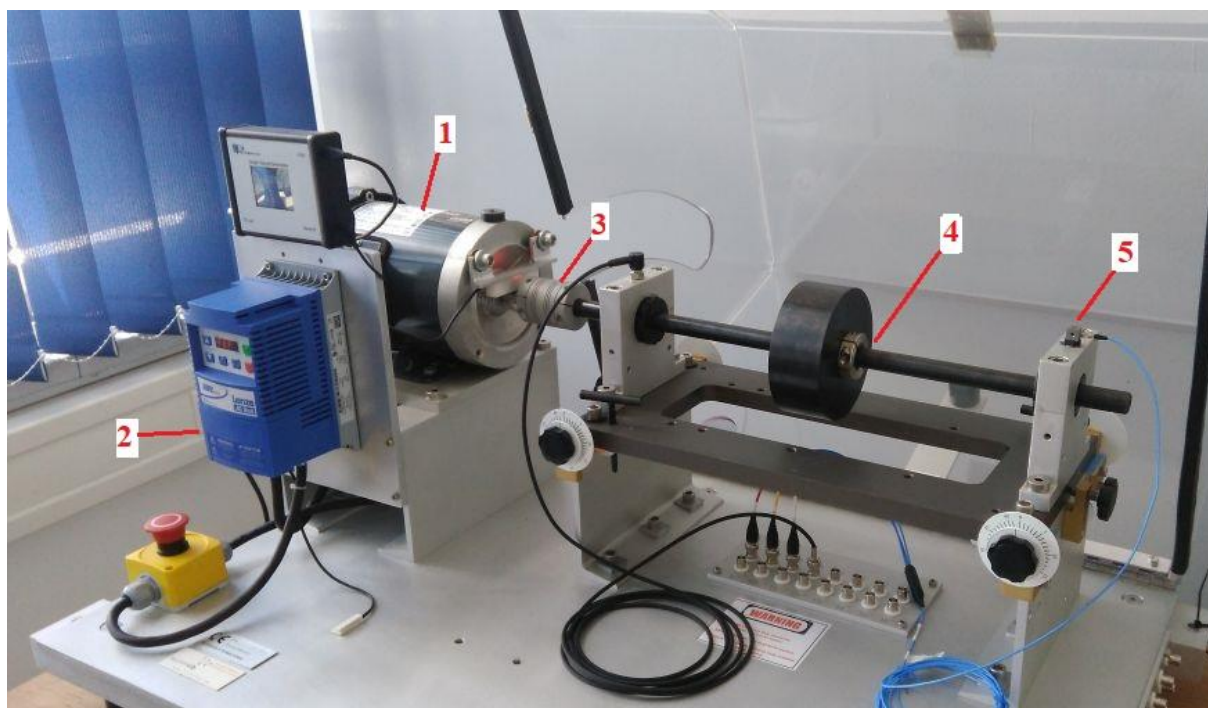
**Tablica 3 Dimenzije sažimajućeg sloja i pripadajućih hiper-parametara**

Na kraju strukture konvolucijske neuronske mreže nalazi se jedan ili više potpuno povezanih slojeva koji su ustvari slojevi iz obične neuronske mreže. U potpuno povezanom sloju nema filtera i mapa značajki, nego se neuroni iz posljednjeg sažimajućeg sloja potpuno povezuju. Kod nadziranih klasifikacijskih zadataka, zadnji sloj mora imati neurona koliko ima i dostupnih klasa, a konvolucijska neuronska mreža završava sa Softmax klasifikacijskom funkcijom koja za izlaz daje vjerojatnost pripadanja testnog primjera određenoj klasi.

#### 4. EKSPERIMENTALNI POSTAV ZA SIMULACIJU KVAROVA ROTACIJSKE OPREME





U sklopu rada bilo je potrebno primjenom senzorskog sustava prikupiti veliku količinu podataka s nekog simuliranog objekta održavanja, iste pripremiti za strojnu obradu te ih iskoristiti za ispitivanje zdravstvenog stanja stroja i dijagnozu kvara pomoću jednog od odabranih algoritama strojnog učenja predstavljenih u poglavlju 3. Za simulaciju objekta održavanja izabran je eksperimentalni postav za simulaciju kvarova rotacijske opreme u Laboratoriju za Održavanje, FSB Zagreb, prikazan na Slika 23. Kako bi se implementirala strategija prediktivnog održavanja po stanju, potrebno je mjeriti odnosno pratiti određeno stanje stroja, a to su u ovom eksperimentalnom pristupu vibracije rotacijske opreme.

Vibracije su mjerene s jednim 1-osnim i 3-osnim akcelerometrom postavljenim na kućišta ležajeva, na računalo snimljeni sustavom za akviziciju podataka National Instruments, koristeći programske pakete LabView te Measurement & Automation Explorer. Za analizu vibracijskih podataka odabran je algoritam dubokog strojnog učenja Konvolucijske neuronske mreže, detaljno opisan u poglavlju 3.5. Algoritam je implementiran u programskog paketu MATLAB, gdje su prethodno prilagođeni podaci sa senzora za ulaz u neuronsku mrežu.



**Slika 23** Simulator kvarova. 1 - Trofazni elektromotor, 2 – Frekventni regulator, 3 – spojka, 4 – modul za simuliranje debalansa (osovina, ležajevi, teret), 5 – 3-osni akcelerometar

Simulator kvarova „SpectraQuest Fault Simulator Expert“ opremljen je trofaznim elektromotorom s frekvencijskim pretvaračem snage 0,75 kW, ugrađenim digitalnim tahometrom i modulom za simuliranje debalansa. Za potrebe eksperimenta, a u cilju simuliranja pojedinih tipova kvarova koristit će se dodatni moduli:

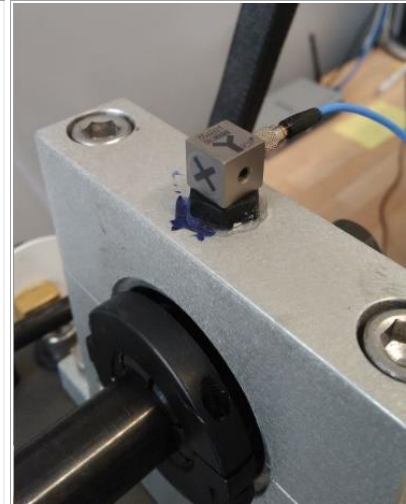
<b>1. Ekscentrični rotor (za vratilo MR-SCK-3/4) (M-ER-3/4)</b>	
<ul style="list-style-type: none"> <li>Asimetričan centar mase</li> </ul>	
<b>2. Nagnuti rotor (za vratilo MR-SCK-3/4) (M-CR-3/4)</b>	
<ul style="list-style-type: none"> <li>0,5 stupnjeva</li> </ul>	
<b>3. Uteg (za vratilo MR-SCK -3/4) (M-BL-3/4)</b>	
<ul style="list-style-type: none"> <li><math>m = 5\text{kg}</math></li> </ul>	
<b>4. Rotor sa simuliranim debalansom</b>	
<ul style="list-style-type: none"> <li>Vijak na rubu rotora unosi debalans</li> </ul>	

**Tablica 4 Moduli za simuliranje kvarova na rotacijskoj opremi**

Pomoću dva akcelerometra mjere se vibracije na rotacijskom uređaju tijekom rada u različitim uvjetima. Na Slika 23 s oznakom 5, označen je 3-osni akcelerometar postavljen na kućište desnog ležaja, dok je na kućište lijevog ležaja postavljen 1-osni akcelerometar koji mjeri vibracije samo u Z osi.

### 1. 3-osni akcelerometar IEPE PCB 356B21

- IEPE
- 1,02 mV/g
- $\pm 500$  g
- Y i Z os 2 – 10 000 Hz, X os 7 000 Hz



### 2. 1-osni akcelerometar IEPE PCB 607A11

- IEPE
- 10,2 mV/(m/s<sup>2</sup>)
- $\pm 50$  g
- 0,5 – 10 000 Hz



**Tablica 5** Mjerni senzori

Podaci se na računalo prebacuju preko sustava za akviziciju podataka:

### *National Instruments NI 9191 + NI 9234 + LabView2014*

- 4-kanalni analogni (+/- 5 V) sustav
- 51,2 kS/s/po kanalu
- LAN
- WLAN
- Prikupljanje podataka kontrolirano programiranim sustavom unutar okruženja LabView 2014



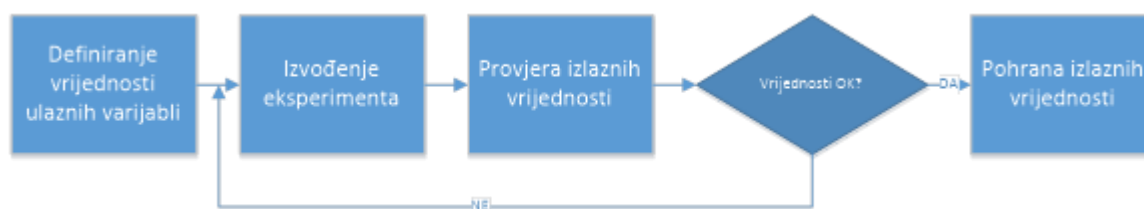
**Tablica 6** Karakteristike sustava za akviziciju podataka

Unutar programskog okruženja LabView 2014 prikazanog na Slika 25 mogu se definirati parametri eksperimenta, prikazani na sljedećoj tablici:

Parametri eksperimenta:
<ul style="list-style-type: none"> <li>• Frekvencija uzorkovanja <math>f_s = 6400 \text{ Hz}</math></li> <li>• Nyquistova frekvencija <math>f_{Ny} = 3200 \text{ Hz}</math></li> <li>• Vrijeme uzorkovanja <math>T_s = 1 \text{ s}</math></li> <li>• Broj uzorkovanja u svakom mjerenju <math>N_s = F_{Ny} * T_s = 3200</math></li> <li>• Prikupljanje podataka kontrolirano programiranim sustavom unutar okruženja LabView 2014</li> <li>• Vrijeme između mjerenja: <math>2 \text{ s}</math></li> <li>• Ukupan broj mjerenja za pojedini set ulaznih parametara: 306</li> <li>• Broj okretaja motora <math>n = 1500 \text{ okretaja/min}</math></li> </ul>

**Tablica 7 Parametri eksperimenta**

Eksperiment se provodi na način definiran dijagramom tijeka na Slika 24 za svaku kombinaciju ulaznih varijabli.



**Slika 24 Dijagram tijeka provođenja eksperimenta prikupljanja vibracija**

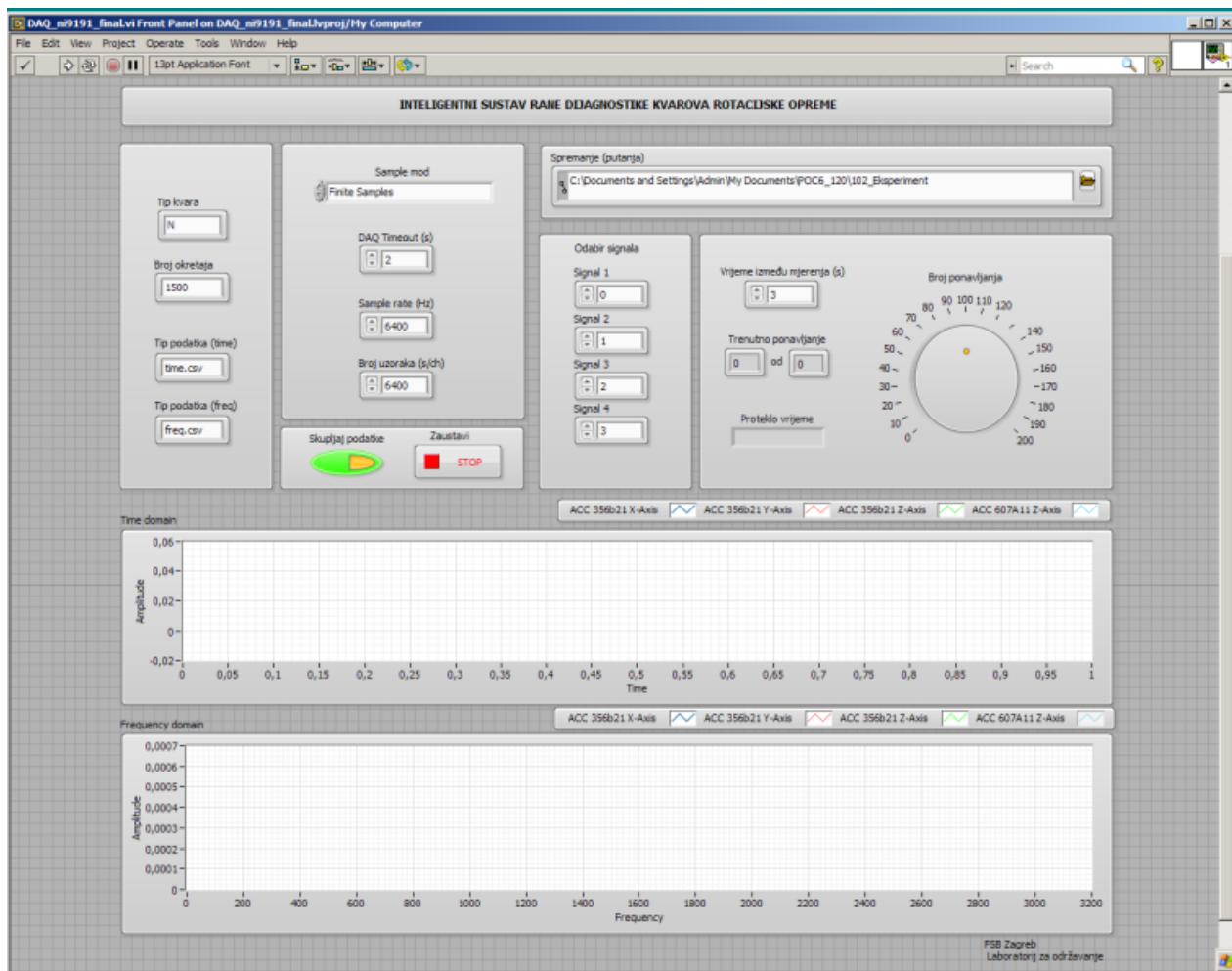
Predviđeni izlaz sastoji se od izmjerenih podataka prikupljenih s tro-osnog akcelerometra u vremenskoj i frekvencijskoj domeni. Izlazni podaci se spremaju tablično u .csv format datoteke. Naziv spremljene datoteke sastoji se od:

1) Podatka o tipu kvara:

- Normal
- Debalans
- Ekscentar
- Nagnuti rotor

2) Podatak o broju okretaja

- a) Broj okretaja\_RPM
- 3) Podatak o vremenskoj ili frekvencijskoj domeni
- a) Time
- b) Freq



**Slika 25** Izgled sučelja za snimanje podataka vremenske i frekvencijske domene na računalo

Zapis u spremljenoj datoteci vremenske domene sadrži sljedeća polja:

- Podatak u vremenu (Timestamp)
- Snaga vibracija (RMS) – X os
- Snaga vibracija (RMS) – Y os
- Snaga vibracija (RMS) – Z os
- Tip kvara, broj okretaja i broj mjerenja
- Snaga vibracija (RMS) – Z1 os (akcelerometar na ležaju sa strane motora)



---

Zapis u spremljenoj datoteci frekvencijske domene sadrži sljedeća polja:

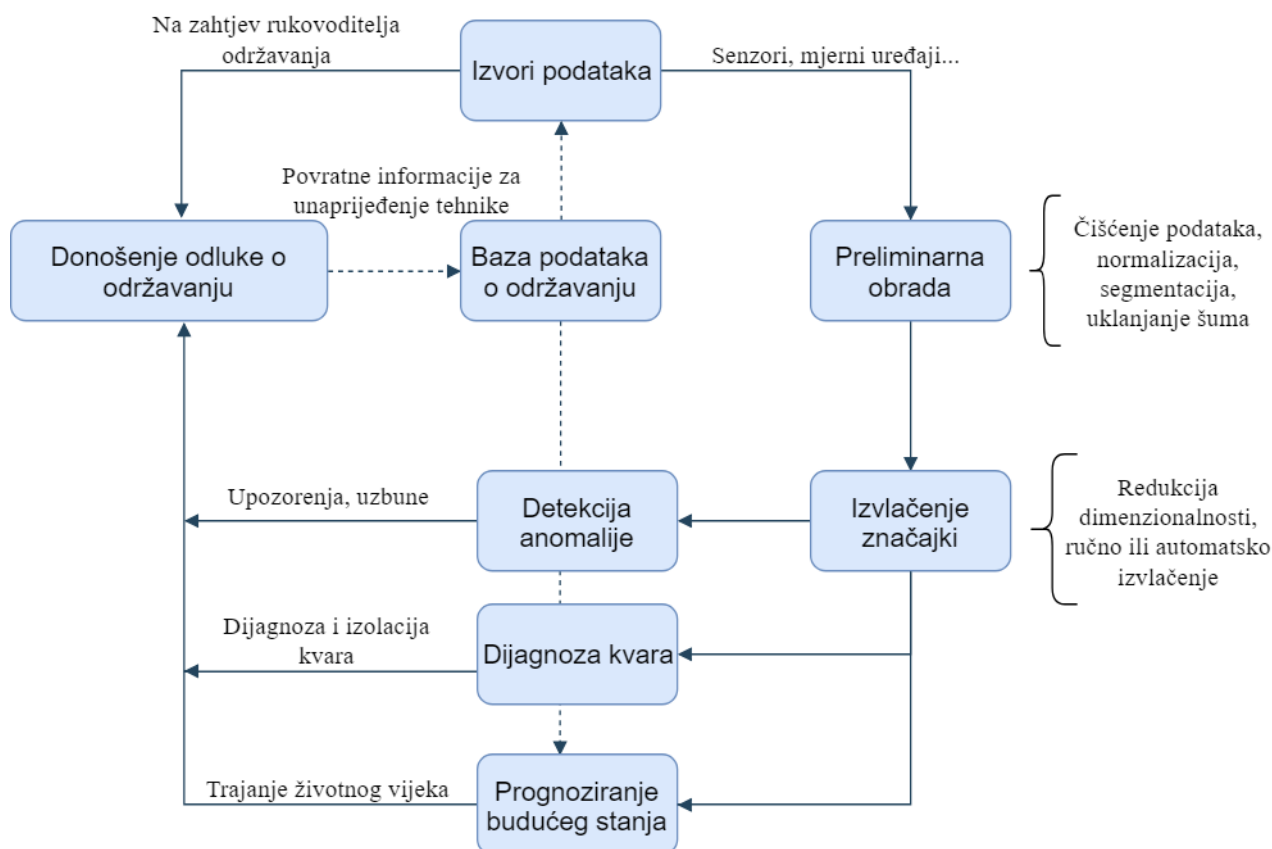
- Trenutna frekvencija
- Snaga vibracija (RMS) – X os
- Snaga vibracija (RMS) – Y os
- Snaga vibracija (RMS) – Z os
- Tip kvara, broj okretaja, broj mjerenja
- Snaga vibracija (RMS) – Z1 os (akcelerometar na drugom ležaju)

Kvarovi simulirani pomoću opreme prikazane u Tablica 4 uvode u sustav različite karakteristične tipove kvarova rotacijske opreme, a koji posljedično stvaraju karakteristične oblike vibracijskog signala.



## 5. INTELIGENTNI SUSTAV PROCJENE ZDRAVSTVENOG STANJA STROJNE OPREME

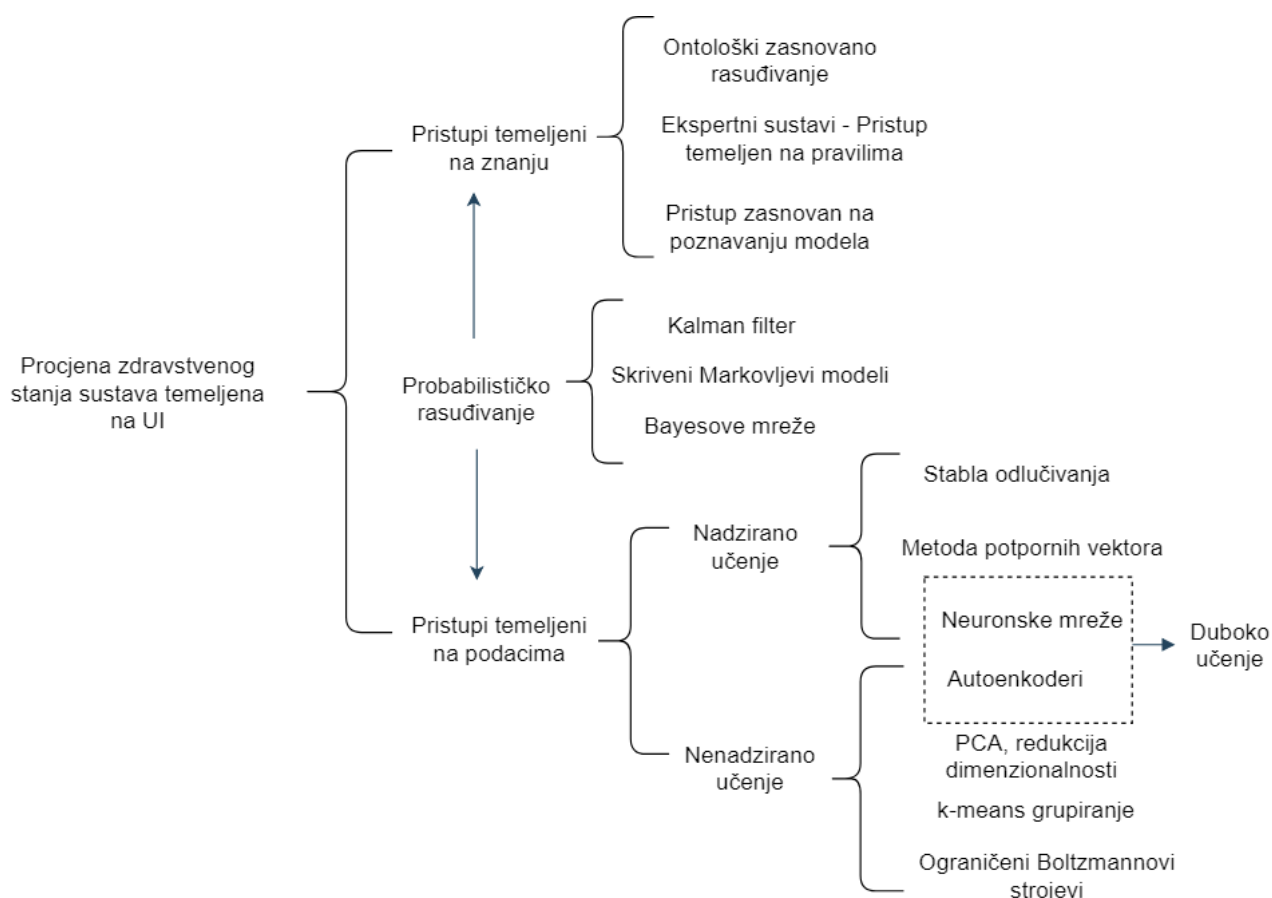
Sustavi procjene i upravljanja zdravstvenog stanja strojne opreme sastoje se od međusobno povezanih postupaka prikupljanja podataka i praćenja stanja opreme te algoritama za detekciju nepredviđenih anomalija, dijagnozu i lokalizaciju kvarova i predviđanje korisnog vijeka trajanja opreme. Potom se informacije iz tih algoritama koriste pri donošenju odluke i izvršavanju operacije održavanja. Nakon izvršene operacije održavanja, svi dobiveni rezultati se pohranjuju u bazu podataka pomoću koje se adaptiraju senzorni mehanizmi i već spomenute aplikacije za detekciju kvarova. Struktura jednog sustava za procjenu i upravljanje zdravstvenog stanja strojne opreme može se prikazati grafički:



Slika 26 Struktura sustava za upravljanje zdravstvenim stanjem strojne opreme. Adaptirano prema [34].

U cilju postizanja što boljih rezultata glede produktivnosti i efikasnosti industrijskih postrojenja, potrebno je smanjiti vrijeme potrebno za obavljanje dijagnoze i popravka kvarova. Tu dolazi do izražaja posjedovanje mnoštva izvora podataka odnosno senzora, koji pružaju raznovrsne informacije o stanju opreme, okolišnim uvjetima i slično. Sa što većom primjenom

senzora u sustavima održavanja industrijske opreme posljedično dolazi do prikupljanja jako velikih količina podataka o stanju opreme. Kada se uzme u obzir ogromna količina novih podataka, povezana s kompleksnošću uslijed različitih mogućih uvjeta rada, može doći do stvaranja većih razina nesigurnosti u procesu dijagnostike zdravstvenog stanja opreme [35]. U tom kontekstu, potrebni su novi pristupi podešavanja aplikacija analize i predviđanja stanja industrijske opreme. Kako bi se riješio taj izazov, konvencionalne tehnike analize podataka zamjenjuju se tehnikama temeljenim na primjeni umjetne inteligencije [24]. Pristupi temeljeni na primjeni umjetne inteligencije mogu se kategorizirati na pristupe temeljene na znanju i temeljene na podacima, a detaljnije su prikazane na Slika 27.

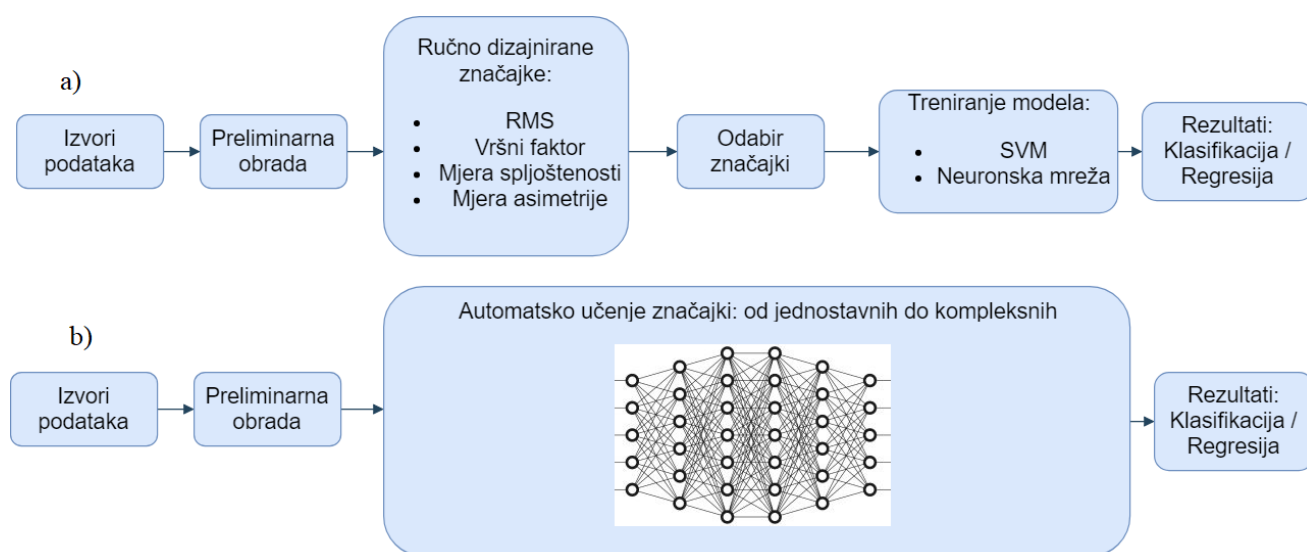


**Slika 27 Kategorizacija metoda procjene zdravstvenog stanja sustava temeljenih na umjetnoj inteligenciji. Adaptirano iz [24].**

Općenito, pristupi temeljeni na znanju i poznavanju modela sustava su vrlo točni i efikasni ako nam je dostupan potpuni matematički opis modela rada mehanizma te sve karakteristične značajke pojedinih modela kvarova i njihovog stupnja progresije. Međutim, može biti jako teško pronaći eksplicitan matematički model sustava ako kompleksnost sustava raste te dolazi do pojava nesigurnosti u procjeni rada. Zbog toga dolazi do pada točnosti i efikasnosti strategije održavanja industrijske opreme. U tom slučaju, tehnike umjetne inteligencije i pristupi

temeljeni isključivo na podacima pokazali su se upotrebljivim jer uspijevaju samostalno naučiti ponašanje sustava iz prikupljenih podataka te time smanjiti pojavu nesigurnosti u procjeni, slučajnih pogrešaka ili nepristranih, sustavnih senzorskih pogrešaka [24].

Na Slici 26 posebno je istaknuto područje tzv. dubokog učenja, koje obuhvaća najčešće algoritme neuronskih mreža i autoenkodera. Algoritmi dubokog učenja omogućavaju modeliranje višedimenzionalnih reprezentacija podataka te klasificiranje ili predviđanje zdravstvenog stanja stroja tako što koriste višeslojne strukture jedinica za obradu informacija (neurona). Duboko učenje zasniva se na činjenici da model kroz višestruke slojeve obrade podataka sam generira korisne značajke iz kompleksnog, višedimenzionalnog signala te se time oslobađa potrebe za prethodnim poznavanjem karakteristika odnosno značajka određenih stanja ili kvarova opreme. Značajke se generiraju polazeći od najjednostavnijih pa svakim idućim slojem stvarajući sve kompleksnije reprezentacije ulaznog signala. Zbog takvog potencijala obrade kompleksnih podataka kao što su podaci o stanju stroja (vibracije, temperatura, tlak...) algoritmi dubokog učenja počinju se primjenjivati na sustave za održavanje i upravljanje zdravstvenim stanjem strojne opreme [24].



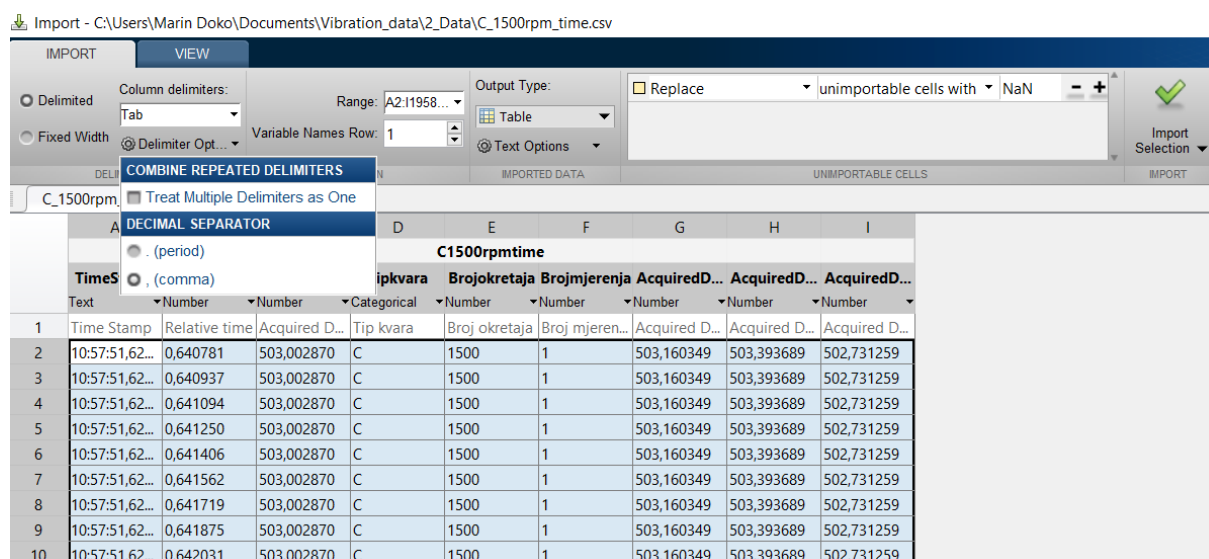
**Slika 28** Razlika između a) konvencionalnih metoda strojnog učenja b) dubokog učenja [24]

Nakon što su istaknute potencijalne prednosti u korištenju algoritama dubokog strojnog učenja za analizu stanja strojne opreme u svrhu dijagnoze kvarova, kao i mogućnosti i prednosti primjene vibracijskih signala u svrhu prediktivnog održavanja u poglavlju 2.3, u sklopu diplomskog rada bit će primijenjen algoritam Konvolucijskih neuronskih mreža (CNN) opisanih u poglavlju 3.5 na vibracijske podatke prikupljene s eksperimentalnog postava detaljno opisanog u poglavlju 0.

## 5.1. Učitavanje snimljenih podataka unutar programskog okruženja MATLAB

Algoritam konvolucijskih neuronskih mreža izrađen je u programskog paketu MATLAB koristeći paket „*Neural network Toolbox*“.

Za početak potrebno je učiniti potrebne preliminarnne operacije obrade snimljenih podataka kako bi konvolucijska neuronska mreža na ispravan način mogla pristupiti podacima za trening te potom i testiranje naučene mreže. Obzirom da su podaci sa eksperimentalnog postava spremljeni u .csv (eng. *Comma separated value*, datoteka s vrijednostima odvojenim zarezom) datoteku kao jedna velika tablica, potrebno je prvotno tablicu učitati u MATLAB, pa potom svako mjerenje razdvojiti na posebnu varijablu i datoteku. Za učitavanje tabličnih datoteka u MATLAB iskorištena je funkcija *Import*, gdje se za decimalno odvajanje mora izabrati zarez (eng. comma) kako bi se vrijednosti iz tablice pravilno očitale unutar varijabli.



Slika 29 Učitavanje tablične datoteke unutar MATLAB-a

Rezultat učitavanja je tablica s izuzeta četiri stupca koja sadržavaju vrijednosti amplituda vibracija u vremenskoj domeni za X, Y, Z os te podatak o rednom broju mjerenja. Za potrebe treninga konvolucijske neuronske mreže, potrebno je svako pojedino mjerenje izdvojiti u posebnu varijablu i spremiti u posebnu .mat datoteku unutar posebne mape. Skripta za preoblikovanje podataka prikazana je u Prilogu.

## 5.2. Odabir hiperparametara i treniranje konvolucijske neuronske mreže

Sljedeći korak je postupak treniranja konvolucijske neuronske mreže na izmjerenim vibracijskim signalima. S obzirom na mogućnost algoritama dubokog strojnog učenja za automatsko učenje i izvlačenje karakterističnih značajki iz vibracijskog signala, ovim eksperimentom pokušat će se dokazati efikasnost konvolucijskih neuronskih mreža u

klasifikaciji različitih vrsta simuliranih kvarova na rotacijskoj opremi direktnom primjenom signala sa senzora vibracija, bez prethodnih ručno generiranih značajki. Zadatak konvolucijske neuronske mreže će biti samostalno učenje značajki vibracijskog signala pomoću kojih će izvesti razlikovanje i klasifikaciju četiri vrste stanja rotacijske opreme kao što je opisano u četvrtom poglavlju:

- Nagnuti rotor
- Ekscentrični rotor
- Rotor sa simuliranim debalansom
- Normalno stanje rotacijske opreme

Prema tome, podaci će biti spremljeni u četiri različite mape, na koje će se pozivati algoritam učenja konvolucijske neuronske mreže.

Prikupljena mjerenja bit će podijeljena na skup trening i testnih primjera kako bi se provjerila generalizacija neuronske mreže na primjerima kojima mreža nije imala pristup tijekom učenja vlastitih parametara. Mjerenja su podijeljena u sljedećem omjeru:

- Trening primjeri: 80% mjerenja
- Testni primjeri: 20% mjerenja

Potom je potrebno izabrati pravu strukturu i hiperparametre konvolucijske neuronske mreže. Način provjere rada i uspješnosti različitih struktura mreže provodi se jednostavnom metodom pokušaja, a za potrebe eksperimenta najbolje rezultate dala je struktura prikazana na Tablica 8. Struktura je sačinjena od dva skupa slojeva gdje svaki skup ima po jedan konvolucijski sloj, aktivacijski sloj i sažimajući sloj. Prvi konvolucijski sloj posjeduje 16 različitih filtera, koji prolazeći svaki zasebno kroz ulazni signal uče karakteristične težine i stvaraju 16 različitih mapa značajki. Veličina sloja je izabrana tako da svojom širinom uhvati kretnje i promjene amplituda vibracija u ulaznom signalu. Izračunate vrijednosti konvolucije se potom transformiraju nelinearnom zglobnom aktivacijskom funkcijom (Slika 12), a onda se te vrijednosti sažimaju metodom maksimalnog sažimanja (poglavlje 3.5).

Ulazni sloj posjeduje tri ulazna kanala zbog tri komponente vibracija po osima. Prvim konvolucijskim slojem, neuronska mreža uči najjednostavnije značajke i uzorke unutar ulaznog signala, a potom se obrađeni signal iz prethodno opisanog sloja šalje u novi skup konvolucijskog, aktivacijskog i sažimajućeg sloja. Ovaj put veličina filtera se smanjuje, dok se broj filtera povećava, kako bi se iz jednostavnih značajki generirale kompleksnije značajke.

Sloj	Veličina sloja i pripadajućih parametara
Ulazni sloj	Ulazni signal: [6400 x 1 x 3]
Konvolucijski sloj 1	16 kernela / filtera veličine: [150 x 1 x 3] Veličina sloja: 6351 x 1 x 16
Aktivacijski sloj 1	ReLU
Sažimajući sloj 1	Maksimalno sažimanje veličine [2 x 1] Veličina sloja: 3176 x 1 x 16
Konvolucijski sloj 2	32 kernela / filtera veličine [5 x 1 x 16] Veličina sloja: 3172 x 1 x 32
Aktivacijski sloj 2	ReLU
Sažimajući sloj 2	Maksimalno sažimanje veličine [2 x 1] Veličina sloja: 1586 x 1 x 32
Potpuno povezani sloj	Veličina sloja: 4
Izlazni sloj	Klasifikacija

**Tablica 8 Odabrana struktura i parametri konvolucijske neuronske mreže**

```

layers =

10x1 Layer array with layers:

   1  'imageinput'    Image Input           6400x1x3 images with 'zerocenter' normalization
   2  'conv_1'       Convolution           16 150x1x3 convolutions with stride [1 1]
   3  'relu_1'      ReLU                  ReLU
   4  'maxpool_1'   Max Pooling           2x1 max pooling with stride [2 2]
   5  'conv_2'       Convolution           32 5x1x16 convolutions with stride [1 1]
   6  'relu_2'      ReLU                  ReLU
   7  'maxpool_2'   Max Pooling           2x1 max pooling with stride [2 2]
   8  'fc'           Fully Connected       4 fully connected layer
   9  'softmax'     Softmax               softmax
  10  'classoutput' Classification Output  crossentropyex with 'Cocked' and 3 other classes

```

**Slika 30 Slojevi mreže unutar MATLAB-a**

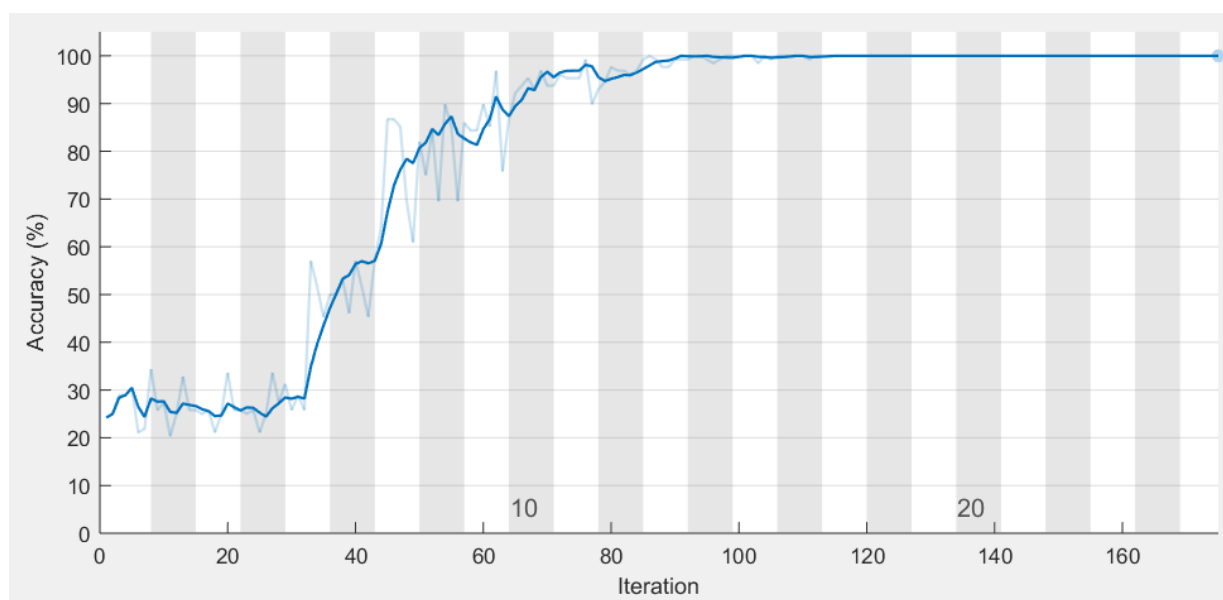
Nakon definiranja slojeva i parametara mreže, potrebno je definirati i metode i parametre učenja. Parametri i način učenja neuronske mreže detaljno je opisan u poglavlju 3.1.2. Kao način učenja mreže odabrana je opisana metoda stohastičkog gradijentnog spuštavanja s uvedenim momentom učenja. Broj maksimalno mogućih epoha učenja može se varirati kako bi se dobio kompromis između efikasnosti i vremena učenja, a tipično se odabire između 20 do 30 epoha učenja. Stopa učenja (Slika 14) mora biti odabrana kao kompromis između sporog učenja i

konvergencije rješenju te bržeg učenja ali moguće divergencije od rješenja. Eksperimentom je utvrđena stopa učenja od 0.007.

Parametri učenja	
Algoritam:	Stohastičko gradijentno spužtanje
Momentum:	DA
Maksimalan broj epoha:	25
Stopa učenja:	0.007

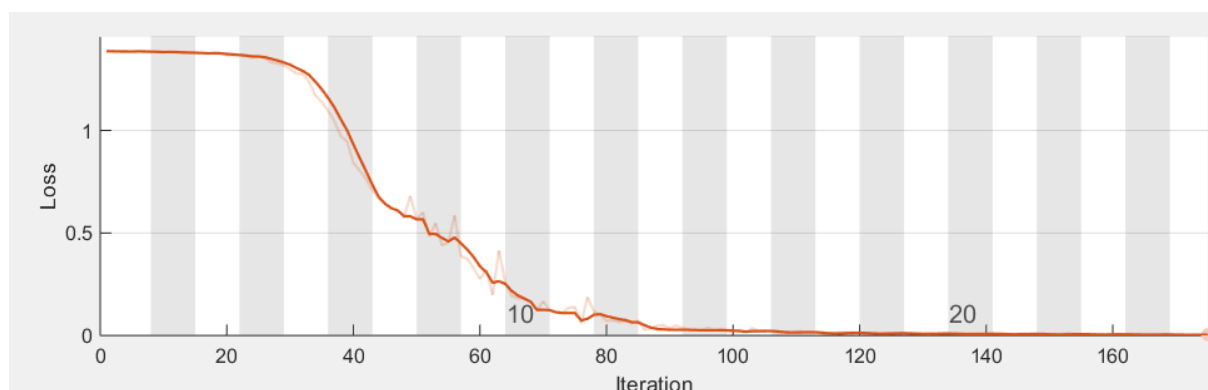
**Tablica 9 Parametri učenja**

Tijek učenja može se prikazati grafički na Slici:



**Slika 31 Točnost klasifikacije tijekom učenja. Epohe su prikazane bijelim i sivim stupcima.**

Osim točnosti klasifikacije, moguće je prikazati i kretanje funkcije gubitka tijekom učenja:



**Slika 32 Funkcija gubitka kroz epohe učenje**

### 5.3. Rezultati testiranja konvolucijske neuronske mreže

Nakon provedenog učenja, potrebno je provjeriti generalizaciju mreže na neviđenim, testnim primjerima. Sljedeći podjelu mjerenja na trening i testne primjere, unutar svake klase stanja opreme ostalo je po 61 mjerenje, što nam daje ukupno 244 testna mjerenja pomoću koje će se utvrditi efikasnost mreže. Točnost mreže računa se po formuli:

$$accuracy = \frac{Broj\ primjera - Broj\ netočnih\ predviđanja}{Broj\ primjera} \quad (28)$$

Nakon obavljenog testiranja točnost klasifikacije mreže je:

$$accuracy = 0.9959$$

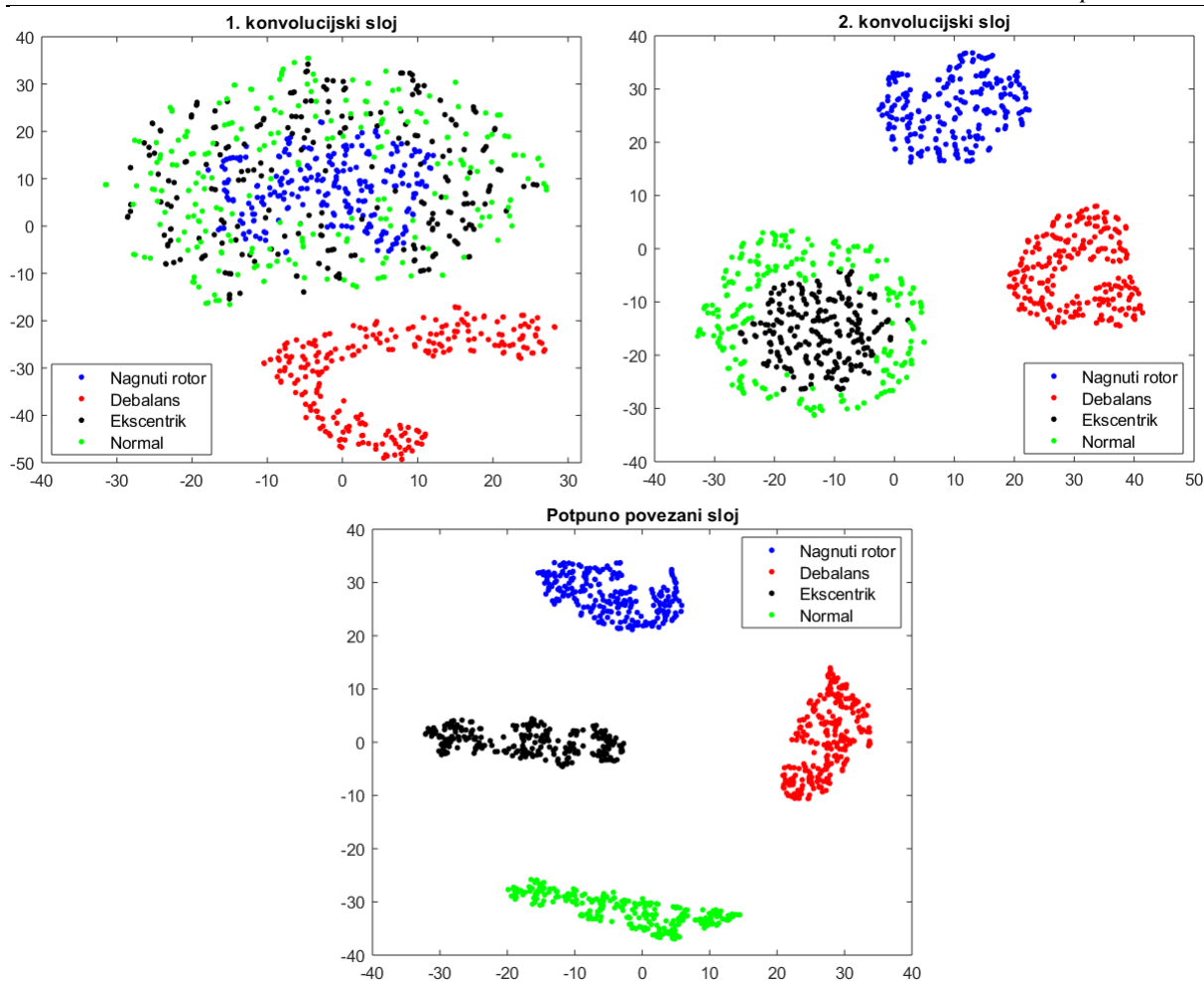
odnosno, mreža nije uspjela točno klasificirati samo jedno mjerenje.

Time je dokazano da je konvolucijska neuronska mreža vrlo uspješno naučila razlikovati različite vrste vibracijskih signala koji su karakteristični za različita loša stanja rotacijske opreme od signala karakterističnog za rad rotacijske opreme u normalnom stanju.

Kako bi se detaljnije i podrobnije opisao i vizualizirao postupak automatskog učenja značajki unutar duboke konvolucijske neuronske mreže, vizualizirat će se grupiranje značajki tijekom slojeva mreže. Značajke vibracijskog signala izvučene su pomoću algoritma t-SNE (*eng. t – Distributed Stochastic Neighbor Embedding*) [36] koji je jedna od varijacija algoritma za redukciju dimenzionalnosti više-dimenzionalnih skupova podataka.

Vidljivo je sa da konvolucijska neuronska mreža kroz slojeve postupno uspijeva razlikovati te grupirati i klasificirati značajke zadanih stanja rotacijske opreme. U prvom konvolucijskom sloju već uspijeva razlikovati signal debalansa od ostalih signala, dok u drugom konvolucijskom sloju jako dobro razlikuje signal nagnutog rotora i debalansa. Odnos između ekscentričnog i normalnog stanja rotacijske opreme je izrazito nelinearan što pokazuju koncentrični oblici iz drugog sloja. Naposljetku u potpuno povezanom sloju, mreža uspješno razlikuje sva četiri zadana stanja opreme.



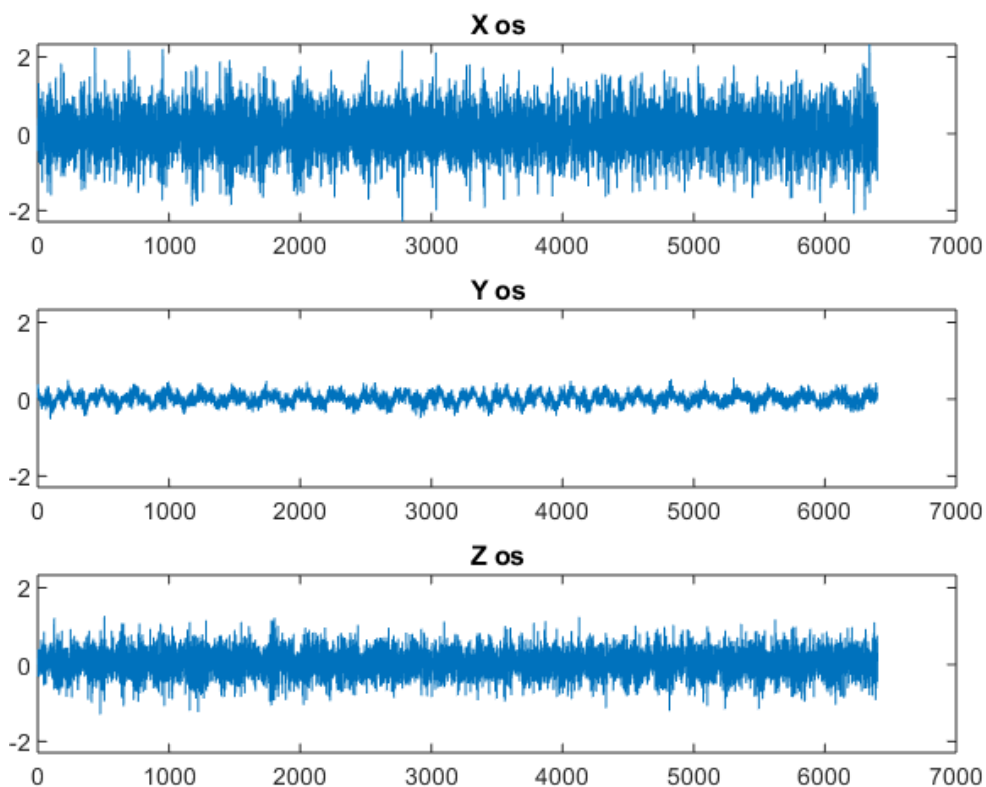


**Slika 33 Vizualizacija automatskog učenja značajki kroz slojeve konvolucijske neuronske mreže pomoću t-SNE algoritma**

Potom će biti vizualno prikazani kerneli ili filteri prvog konvolucijskog sloja. Pomoću tih naučenih kernela, konvolucijska neuronska mreža obrađuje svaki novi, testni signal koji se pošalje u mrežu i pomoću tih kernela uspijeva izlučiti značajke vizualizirane na prethodnoj slici. Veličina kernela i sloja na koji djeluju prikazna je na Tablica 8.

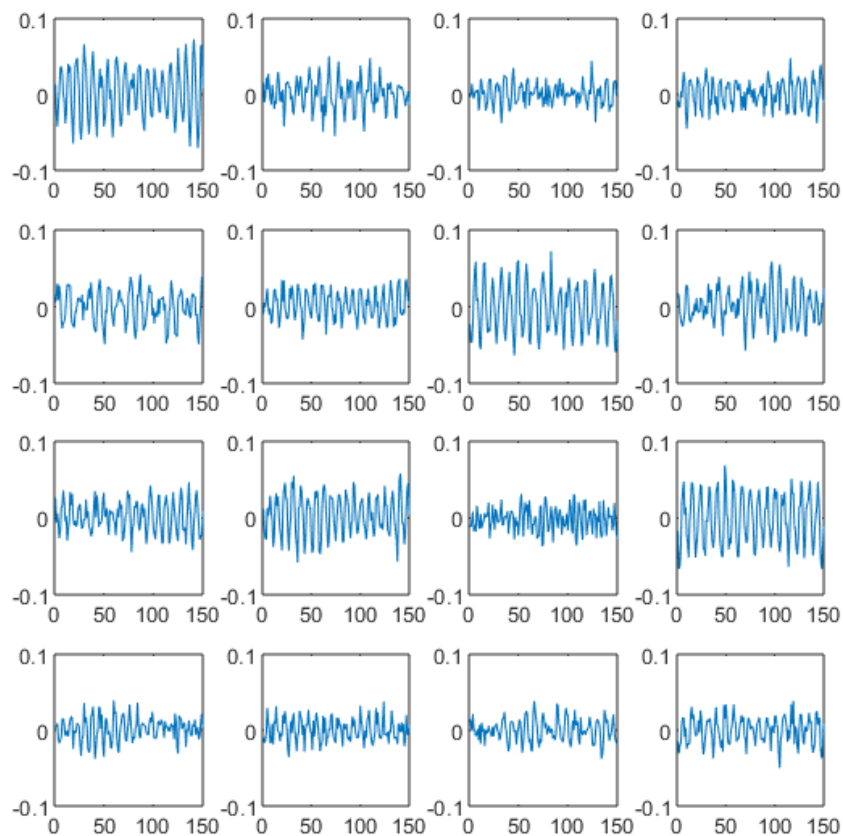
Za bolje razumijevanje bit će prikazan jedan primjer prikupljenog vibracijskog signala ekscentričnog rotora u sve tri izmjerene osi. Na Slika 34 vidljivo je očitavanje najvećih amplituda vibracija na X osi, malo manjih amplituda na Z osi, te minimalnih amplituda na Y (aksijalnoj) osi. Prema tome prikazivanje filtera ograničit će se na os gdje se pojavljuju najveće vibracije koje neuronska mreža mora prepoznati i izvući značajke o toj vrsti vibracija. Kerneli za X os u vremenskoj domeni prikazani su na Slika 35. Vidljivi su različiti oblici kernela koji obrađuju ulazni signal u potrazi za najjednostavnijim karakterističnim značajkama. Značajke vibracijskog signala se još bolje mogu prikazati prelaskom u frekvencijsku domenu, pa se tako

na Sliku 36 prikazuju kerneli iz prethodne slike prebačeni u frekvencijsku domenu direktnom Fourierovom transformacijom.

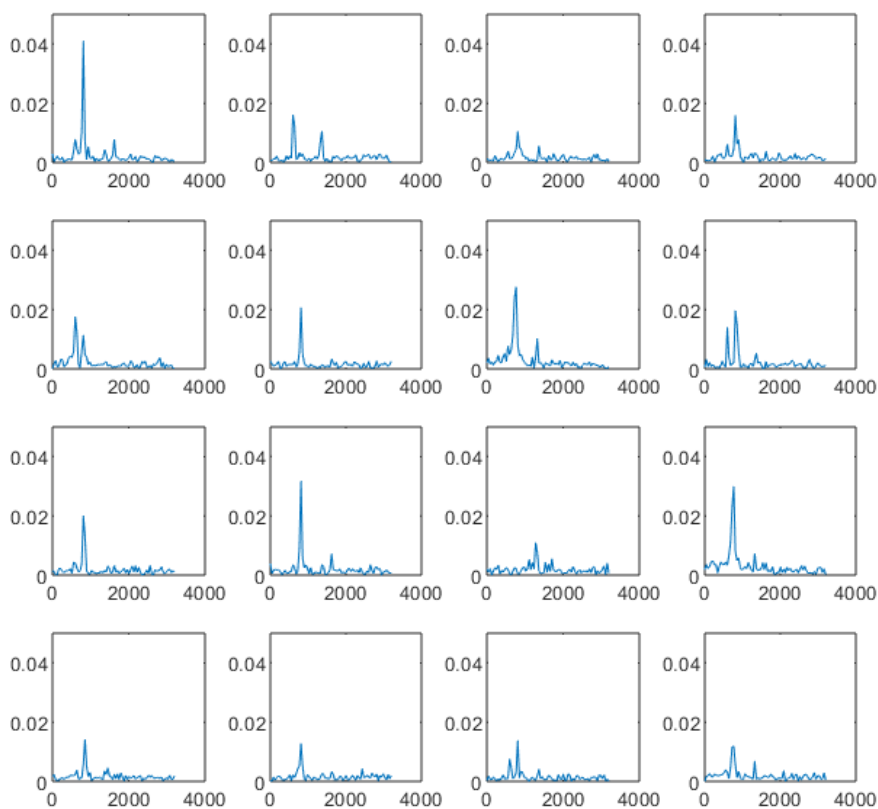


**Slika 34** Primjer izmjerenih vibracija jednog mjerenja ekscentričnog rotora

Pomoću Slika 36 može se zaključiti da kerneli u prvom konvolucijskom sloju pretražuju i očitavaju jednostavne nisko-frekvencijske ili srednje-frekvencijske amplitude vibracijskog signala. Svaki zasebni kernel uglavnom očitava jednu frekvencijsku komponentu da bi se kombiniranjem svih značajki iz prvog sloja u idućem sloju pronalazile kompleksnije značajke odnosno kombinacije različitih specifičnih amplituda frekvencija koje se pojavljuju u vibracijskom signalu. Postupnim prepoznavanjem značajki algoritam konvolucijskih neuronskih mreža omogućuje postizanje odličnih rezultata u raspoznavanju loših stanja strojne opreme od normalnog stanja rada.



**Slika 35** Prikaz kernela prvog konvolucijskog sloja u vremenskoj domeni



**Slika 36** Prikaz kernela prvog konvolucijskog sloja u frekvencijskoj domeni

## 6. ZAKLJUČAK

Unutar diplomskog rada prikazane su mogućnosti i prednosti primjene algoritama strojnog učenja, točnije algoritma konvolucijske neuronske mreže, na podacima prikupljenim na simulatoru kvarova rotacijske opreme, a u svrhu prediktivnog održavanja po stanju. Izmjerene amplitude vibracija na rotirajućoj strojnoj opremi daju najviše mogućih informacija o trenutnom zdravstvenom stanju opreme te su one izabrane kao podaci pomoću kojih će se vršiti dijagnostika odnosno predikcija kvarova na opremi. Primjenom tehnologije dubokih neuronskih mreža moguće je reducirati vrijeme potrebno za ručno izvlačenje karakterističnih značajki iz signala, koji dolazi sa senzora vibracija te time poboljšati efikasnost algoritma i ubrzati zadatke dijagnostike i predikcije stanja. Višeslojna struktura konvolucijskih neuronskih mreža uspijeva prepoznati i izvući kompleksne i nelinearne značajke iz vibracijskog signala, počevši od najjednostavnijih u prvom sloju, sve do kompleksnih značajki u idućim slojevima. Na kraju, prema provedenom eksperimentu, moguće je s velikom točnošću od 99,6% izvršiti klasifikaciju različitih stanja strojne opreme pomoću automatski izvučenih značajki. Naučene značajke se iz konvolucijske neuronske mreže mogu koristiti i unutar drugih algoritama strojnog učenja.

Daljnji smjer istraživanja može uključivati dodatno istraživanje mogućnosti primjene „inteligentnih“ algoritama dubokog strojnog učenja unutar prediktivnog održavanja strojeva, kako bi se povećala sigurnost rada takvih sustava i omogućila primjena na realnim industrijskim pogonima. Specifično, primarno se definira potreba za primjenom algoritama nenadziranog strojnog učenja u području otkrivanja anomalija kod sustava koji su već u radu, a kod kojih ne postoji dovoljno znanje o mogućim tipovima kvarova. Primjenom takvih algoritama omogućilo bi se prediktivnom sustavu da automatski prepozna signale, koji ukazuju na anomalije te bi se onda takve značajke implementirale unutar opisanog sustava konvolucijske neuronske mreže za dijagnostiku, predikciju i lokalizaciju kvarova.

**LITERATURA**

- [1] R. Keith Mobley: „An introduction to predictive maintenance“
- [2] R. Kothmasu, S.H. Huang, W.H. VerDuin: „System Health Monitoring and Prognostics – A Review of Current Paradigms and Practice“; Handbook of Maintenance Management and Engineering, Springer London, 2009.
- [3] D. Lisjak: „Bilješke s predavanja kolegija: Održavanje“
- [4] G. P. Sullivan: „Operations and Maintenance best practice“; US Department of Energy, 2010.
- [5] R. B. Randall: „Vibration-based Condition monitoring. Industrial, Aerospace and Automotive applications“; Wiley, 2011.
- [6] Jardine, A. et al: „A review on machinery diagnostics and prognostics implementing condition-based maintenance“, Mechanical Systems and Signal Processing 2006; 20: 1483-1510
- [7] Lee, I., Lee, K.: „The Internet of Things (IoT): Applications, investments and challenges for enterprises“, 2015., Kelley school of Business, Indiana University
- [8] Xu, L.,D., He, W., Li., S: „Internet of Things in Industry: A Survey“ 2014., IEEE Transactions on Industrial Informatics vol. 10, no. 4, pp. 2233-2243
- [9] Haarman, M. et al. „Report: Predictive Maintenance 4.0, Predict the unpredictable“ 2017., Mainnovation
- [10] Rübmann, M., et al.: „Industry 4.0: The future of productivity and growth in manufacturing industries.“, 2015., Boston Consulting
- [11] Chesworth, D., „Industry 4.0 Techniques as a Maintenance Strategy (A review paper)“, 2018.
- [12] Mitchell, T. : „Machine Learning“, 1998.
- [13] GoodFellow, I., et al.: „Deep learning“, 2016., MIT Press; [www.deeplearningbook.org](http://www.deeplearningbook.org)
- [14] Taigman, Y., et al.: „DeepFace: Closing the Gap to Human-Level Performance in Face Verification“, 2014, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- [15] Chandola, V., Banerjee, A., and Kumar, V.: „Anomaly detection: A survey“, 2009.. ACM Comput. Surv. 41, 3, Article 15, 58 pages
- [16] Bishop, C.,M. : „Pattern Recognition and Machine Learning“, 2006., Springer US
- [17] Rosenblatt, F., „The Perceptron: A probabilistic model for information storage and organization in brain“, 1958., Psychological review, Vol. 65, No. 6.

- [18] CS231n: Convolutional Neural Networks for Visual Recognition – Stanford Course:  
<http://cs231n.stanford.edu/>
- [19] Bašić, B., D., Čupić, M., Šnajder, J., „Umjetne neuronske mreže“, 2008., FER Zagreb
- [20] Rumelhart, D., E., Hinton, G., E., Williams, R., J., „Learning representations by back-propagation error“, 1986., Nature, 323: 533-536
- [21]
- [22] Wang J., et al. : „Deep learning for smart manufacturing: Methods and Applications“, 2018., J. Manuf. Syst.
- [23] <https://deeplearning4j.org/lstm.html#two>
- [24] Khan, S., Yairi, T., : „A review on the application of deep learning in system health management“, 2017., Mechanical System and Signal Processing 107: 241-265
- [25] Stamenković, F. „Duboke neuronske mreže – seminarski rad“; FER, Zagreb
- [26] Cybenko, G: „Approximations by Superpositions of a Sigmoidal Function“. Math. Control Signals System 1989; 2: 303-314.
- [27] Hornik, K: „Multilayer Feedforward networks are universal approximators“ Neural Networks 1989; 2: 359-366.
- [28] LeCunn, Y., et al.: „Gradient-based learning applied to document recognition“. Proc of the IEEE, 1998.
- [29] Krizhevsky, A., Sutskever, I. & Hinton, G: „ImageNet classification with deep convolutional neural networks“. Advances in Neural Information Processing Systems 2012; 25: 1090–1098.
- [30] Hinton, G. et al. : „Deep neural networks for acoustic modeling in speech recognition“. IEEE Signal Processing Magazine 2012; 29: 82–97.
- [31] Helmstaedter, M. et al.: „Connectomic reconstruction of the inner plexiform layer in the mouse retina“ 2013; Nature 500, 168–174
- [32] Xiong, H. Y. et al. „The human splicing code reveals new insights into the genetic determinants of disease“. 2015; Science 347, 6218
- [33] Michael A. Nielsen, „Neural Networks and Deep Learning“, Determination Press, 2015
- [34] P. Bonissone, X. Hu, R. Subbu, „A systematic PHM approach for anomaly resolution: A hybrid neural fuzzy system for model construction“, 2009., Proc, PHM.
- [35] S. Khan, P. Phillips, C. Hockley, I. Jennions, : „No Fault Found events in maintenance engineering Part 2: Root causes, technical developments and future research“, Reliab. Eng. Syst. Safety 123 (2014) 196–208
- [36] <https://lvdmaaten.github.io/tsne/>