Procjena kvarova ležajeva primjenom strojnog učenja

Biluš, Luka

Master's thesis / Diplomski rad

2019

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:776526

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb







SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Luka Biluš

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor: Student:

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

Zagreb, 2019.

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

Zahvaljujem se 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,

Naposljetku, zahvaljujem se svojoj obitelji čija mi je podrška i razumijevanje tijekom svih godina studiranja, dala snage za uspješni dovršetak studija.

Luka Biluš

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: **LUKA BILUŠ** Mat. br.: 0035191738

Naslov rada na hrvatskom jeziku: **PROCJENA KVAROVA LEŽAJEVA PRIMJENOM STROJNOG UČENJA**

Naslov rada na engleskom jeziku: **MACHINE LEARNING BASED BEARING FAULT DIAGNOSIS**

Opis zadatka:

Ležajevi su elementi strojeva koji služe za vođenje pokretnih strojnih dijelova te prenose vanjska opterećenja između tih dijelova i onih koji se nalaze u relativnom gibanju prema njima. Ispravan rad ležaja često je od presudnog značenja za integritet i vijek trajanja strojeva, a prema istraživanjima, svega 10 do 20% ležajeva doživi svoj predviđeni vijek trajanja. Razlozi tome uglavnom su posljedica neispravnog podmazivanja, zamora materijala, nepravilne montaže, kontaminacije i sl. Tijekom povijesti održavanja razvijene su brojne tehnike održavanja ležajeva, od čega danas najznačaniju ulogu u održavanju ima dijagnostika stanja ležajeva na temelju vibracijskih podataka. Strojno učenje sve je prisutnija tehnika obrade velike količine podataka, a umjetne neuronske mreže sve šire prihvaćen alat računalne inteligencije. Sukladno navedenome, ovim radom potrebno je:

1. Opisati metodologiju dubokog strojnog učenja s težištem na primjenu konvolucijskih neuronskih mreža.
2. Definirati proces održavanja po stanju primjenom strojnog učenja.
3. Analizirati dostupne podatke o vibracijskom stanju kugličnih ležajeva.
4. Kreirati računalni model procjene kvarova ležajeva primjenom strojnog učenja.
5. Na temelju rezultata modela, dati zaključak.

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

Zadatak zadan:
15. studenog 2018.

Rok predaje rada:
17. siječnja 2019.

Predviđeni datum obrane:
23. siječnja 2019.
24. siječnja 2019.
25. siječnja 2019.

Zadatak zadao:

prof. dr. sc. Dragutin Lisjak

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

SADRŽAJ

1. UVOD 1
2. STRATEGIJE ODRŽAVANJA 2
2.1. Korektivno održavanje 5
2.2. Preventivno održavanje 6
2.3. Suvremeni koncepti održavanja 8
2.3.1. Plansko održavanje 8
2.3.2. Logistički koncept održavanja 10
2.3.3. Terotehnološki pristup održavanju 10
2.3.4. Ekspertni sustavi 11
2.3.5. Samoodržavanje 14
2.4. Održavanje po stanju (prediktivno održavanje) 16
2.5. Mjerenje i analiza vibracija 18
3. STROJNO UČENJE 21
3.1. Nadzirano učenje 22
3.1.1. Klasifikacija 23
3.1.2. Regresija 23
3.1.3. Logičke relacije 24
3.2. Nenadzirano učenje 25
3.3. Polu-nadzirano učenje 25
3.4. Duboko učenje 26
4. UMJETNE NEURONSKE MREŽE 27
4.1. Model umjetne neuronske mreže 29
4.2. Aktivacijske funkcije 32
4.3. Učenje neuronskih mreža 35
4.3.1. Gradijentni spust 35
4.3.2. Stohastički gradijentni spust 36
4.3.3. Algoritam unazadne propagacije 37
4.4. Regularizacija neuronskih mreža 39
4.5. Konvolucijske neuronske mreže 41
5. PROCJENA UZROKA I INTENZITETA OŠTEĆENJA LEŽAJEVA 48
5.1. Ulazni set podataka 50

5.2. Programsko sučelje 52
5.3. Učitavanje ulaznih podataka 53
5.4. Konvolucijska neuronska mreža i odabir hiperparametara 54
5.5. Rezultati konvolucijske neuronske mreže 58
6. ZAKLJUČAK 62
LITERATURA 63

POPIS SLIKA

Slika 1. Razvoj koncepcija održavanja [6] 4
Slika 2. Krivulja "kade“ [5] 7
Slika 3. Plansko održavanje [7] 9
Slika 4. Elementi ekspertnog sustava [7] 12
Slika 5. Arhitektura sustava samoodržavanja [11] 14
Slika 6. Algoritam zaključivanja kod sustava samoodržavanja [11] 15
Slika 7. Koncept održavanja po stanju [13] 17
Slika 8. Elementi kompresijskog piezoelektičnog akcelerometra [16] 19
Slika 9. Odnos vremenske i frekvencijske domene [4] 20
Slika 10. Raščlamba Strojnog učenja [19] 22
Slika 11. Klasifikacija [19] 23
Slika 12. Regresija [19] 24
Slika 13. Razlika između klasifikacije i regresije [21] 24
Slika 14. Grupiranje (k-means algoritam) [23] 25
Slika 15. Neuron [27] 27
Slika 16. Jednostavni umjetni neuron - perceptron [27] 28
Slika 17. Struktura neuronske mreže [29] 30
Slika 18. Binarna (lijevo) i Bipolarna (desno) funkcija praga 32
Slika 19. Djelomično linearna aktivacijska funkcija 33
Slika 20. Sigmoidalna funkcija [31] 33
Slika 21. ReLU aktivacijska funkcija [31] 34
Slika 22. Veliki korak (lijevo), mali korak (desno) [35] 35
Slika 23. Utjecaj podnaučenosti i prenaučenosti na sposobnost generalizacije [39] 40
Slika 24. Konvolucijska operacija sa receptivnim poljem 5x5 [38] 43
Slika 25. Skriveni sloj sa 3 mapi značajki [38] 44
Slika 26. Postupak sažimanja (maksimalno sažimanje) [43] 44
Slika 27. Primjer strukture konvolucijske neuronske mreže [44] 46
Slika 28. Postupak određivanja stanja opreme izvlačenjem podataka i algoritmom strojnog učenja [45] 48
Slika 29. Razlika između izvlačenja značajki i učenja značajki [45] 49
Slika 30. Ispitna stanica [47] 50

Slika 31. Učitavanje podataka unutar MATLAB-a 54
Slika 32. Slojevi konvolucijske mreže 55
Slika 33. Proces učenja konvolucijske neuronske mreže 57
Slika 34. Funkcija gubitaka tijekom učenja konvolucijske neuronske mreže 57
Slika 35. Vizualizacija značajki na ulaznom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma 59
Slika 36. Vizualizacija značajki u prvom konvolucijskom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma 60
Slika 37. Vizualizacija značajki u drugom konvolucijskom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma 60
Slika 38. Vizualizacija značajki u potpuno povezanom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma 61
Slika 39. Vizualizacija značajki u potpuno povezanom sloju konvolucijske mreže uz pomoć t-SNE algoritma između najuspješnijeg slučaja (lijevo) i najneuspješnijeg slučaja (desno) 61

POPIS TABLICA

Tablica 1. Parametri konvolucijskog sloja 46
Tablica 2. Parametri sloja sažimanja 47
Tablica 3. Specifikacije ležaja na pogonskoj strani motora 51
Tablica 4. Specifikacije ležaja na poziciji ventilatora 51
Tablica 5. Struktura konvolucijske neuronske mreže i njezini parametri 56
Tablica 6. Parametri učenja konvolucijske neuronske mreže 56
Tablica 7. Rezultati učenja konvolucijske neuronske mreže 58

POPIS OZNAKA

Oznaka	Jedinica	Opis
a_j	-	Aktivacija
a^l_j	-	aktivacija *j*-tog neurona u *l*-tom sloju
b	-	Prag
b^l_j	-	Prag u *j*-tom neuronu u sloju *l*
D_n	-	Dubina ulaznog sloja
E	-	Funkcija gubitaka
E_0	-	Neregularizirana funkcija gubitaka
$\nabla E(w)$	-	Gradijentni vektor funkcije gubitaka
F	-	Veličina filtera
f	-	Prijenosna funkcija
H_n	-	Visina ulaznog sloja
h_t	-	Skriveno stanje u trenutku *t*
$h'(z^L_j)$	-	Mjera koliko se brzo aktivacijska funkcija mijenja u z^L_j
K	-	Broj filtera
L	-	Broj slojeva
m	-	Prosječna vrijednost podatka
net	-	Suma težina i ulaznih vrijednosti
P	-	Nadopuna
S	-	Korak filtera
s	-	Standardna devijacija
$t_{\theta l}(.)$	-	Transformacija neobrađenih podataka
U	-	Matrica tranzicije
W_n	-	Širina ulaznog sloja
w_n	-	Težine veza između neurona
w_{ji}	-	Težina između *i*-tog i *j*-tog neurona
w_{j0}	-	Prag
w^l_{jk}	-	težina za vezu *k*-tog neurona u *(l-1)* sloju prema *j*-tom neuronu u sloju *l*
X	-	Ulazne varijable
X_i	-	Ulazi u neuron
Y	-	Izlazne varijable

y	-	Izlazna vrijednost mreže
$\hat{Y}$	-	Predviđanje
z	-	Transformacija nelinearnim aktivacijskim funkcijama
δ	-	Vrijednost greške
δ^L		greška u izlaznom sloju
η	-	Stopa učenja
θ	-	Prag
λ	-	parametar regularizacije
Φ	-	Značajke

SAŽETAK

Održavanje tehničkih sustava u funkcionalnom stanju jedno je od najznačajnijih djelatnosti suvremenih poduzeća. Samim time razvoj održavanja također ima velik značaj, gdje se počinju primjenjivati inteligentni sustavi kojima je moguća rana dijagnostika kvarova unutar sustava održavanja. Primjena algoritama strojnog učenja unutar održavanja po stanju rezultira smanjenjem vremena zastoja, povećanjem kvalitete proizvoda i usluga, omogućavanjem predviđanja kvarova. Unutar ovog rada primijenjen je jedan od algoritama konvolucijskih neuronskih mreža koje spadaju pod strojno učenje. Algoritmom se provela analiza vibracijskih signala ležajeva sa različitim oštećenjima sa ciljem raspoznavanja tih oštećenja bez ikakvog ručnog izvlačenja značajki iz podataka. Vibracijski signali prolaze kroz višeslojnu konvolucijsku mrežu koja automatski uči značajke signala. Točnost klasifikacije ovog algoritma iznosi oko 97,58%.

Ključne riječi: Održavanje po stanju, prediktivno održavanje, vibracije, strojno učenje, konvolucijske neuronske mreže

SUMMARY

The maintenance of technical systems in a functional state is one of the most important activities of modern companies. As a result of that, maintenance development also has great significance, where intelligent systems are beginning to be used to provide early diagnosis of failuers within the maintenance systems. The application of machine learning algorithms within the condition based maintenance results in reduction of downtime, increasing the quality of products and services and enables the prediction of failures. In this thesis, one of the algorithms of convolutional neural networks that are part of machine learning has been applied. The algorithm was used to conduct an analysis of vibration signals of bearings with various damages with the purpose of recognizing those damages without extracting any features from the data. Vibration signals pass through a multilayer convolutional network that automatically learns the signals features. The achived accuracy of the classification with the use of this algorithm is approximately 97,58%.

Key words: Condition based maintenance, predictive maintenance, vibrations, machine learning, convolutional neural networks

1. UVOD

Kvarovi industrijskih strojeva odnosno zastoj u njihovom radu mogu uzrokovati velike ekonomske gubitke, a ponekad predstavljaju i opasnost ljudima koji rade sa tim strojevima. Također, održavanje takvih sustava može činiti ključnu razliku u uspješnosti poduzeća s obzirom da održavanje ima utjecaj na ukupne troškove poduzeća pa i kvalitetu samog proizvoda. Stoga, kako bi se zadržala ispravnost industrijskih strojeva, zahtijevaju se bolje i inteligentnije tehnike praćenja stanja sustava kojima je u cilju povećati vijek trajanja sustava, smanjiti vrijeme zastoja i poboljšati kvalitetu proizvoda. Iako se umjetna inteligencija uglavnom povezuje sa humanoidnim robotima, oblik umjetne inteligencije temeljen na strojnom učenju poprima sve veću primjenu i u modernoj industriji. Sve je veća primjena prediktivnog održavanja sa integriranim raznim metodama umjetne inteligencije koja uključuje hardverska i programska sučelja, a predstavlja održavanje čiji je naglasak na predviđanju pojave kvara, njegovoj dijagnostici i lociranju te predviđanje preostalog iskoristivog vijeka sustava.

Unutar ovog rada u drugom poglavlju opisat će se strategije održavanja strojne opreme, od najstarijih do modernijih te će se opisati tehnika mjerenja i analize vibracija kao jedna od metoda održavanja po stanju koja će se ujedno i koristiti kako bi se došlo do rezultata mjerenja za praktični dio rada.

U trećem poglavlju bit će dana kratka povijest strojnog učenja te će se spomenuti podjela strojnog učenja uz kratke opise tih podjela uz naglasak na duboko učenje.

U četvrtom poglavlju detaljnije će se definirati algoritmi neuronskih mreža i konvolucijska neuronska mreža, koji će biti primijenjeni u praktičnom dijelu rada. Opisat će se njihova povijest, struktura i način na koji rade kako bi došli do ciljanih rezultata.

Peto poglavlje predstavlja praktični dio rada. U ovom poglavlju objašnjen je postupak primjene algoritma konvolucijskih neuronskih mreža u procjeni kvarova ležajeva uz rezultate njegovog rada te su dane informacije o podacima koji će se koristiti za algoritam.

2. STRATEGIJE ODRŽAVANJA

Postoje mnogobrojne definicije održavanja kao i mnogobrojni pojmovi u održavanju koji se različito tumače i prihvaćaju. EFNMS (European Federation of National Maintenance Societies) definira održavanje opreme kao: Održavanje je funkcija poduzeća kojoj su povjerene stalna kontrola nad postrojenjima i obavljanje određenih popravaka i revizija, čime se omogućava stalna funkcionalna sposobnost i očuvanje proizvodnih i pomoćnih postrojenja te ostale opreme [1]. Prema DIN 31051, održavanje se definira kao: Mjera za očuvanje i ponovno uspostavljanje nazivnog stanja te za utvrđivanje i procjenjivanje stvarnog stanja sredstava rada - radnog sustava [2]. Definicija prema Europskoj organizaciji za upravljanje kakvoćom EOQC (European Organization for Quality Control): Održavanje je kombinacija svih tehničkih i odgovarajućih administrativnih aktivnosti predviđenih za očuvanje nekog sredstva rada - radnog sustava ili dovođenje istog u stanje u kojem on može obavljati predviđenu funkciju [3]. Razmatrajući sve ove definicije može se zaključiti da je održavanje skup aktivnosti administrativnog, organizacijskog, tehničkog i tehnološkog karaktera čiji cilj je očuvanje i poboljšanje radnih karakteristika, odnosno osiguranje stanja održavanog sustava koji ima sposobnost obavljanja namjenske funkcije.

Održavanje kao funkcija danas zauzima značajno mjesto u svakom poslovnom sustavu. U prošlosti održavanje je bilo tretirano kao „nužno zlo“ čija se učinkovitost na poslovne sustave smatrala beznačajnom. U ranijim danima industrijske revolucije, strojevi nisu bili pretjerano kompleksni, samim time manje su se i kvarili te nije bilo potrebne za naprednim tehnikama održavanja. Međutim, dolaskom drugog i trećeg vala industrijske revolucije, odnosno dolaskom noviteta poput montažne linije i brze automatizacije, održavanje poprima sve veći značaj. Došlo je do sve manjeg broja ručnog rada i sve veće automatizacije kroz složene strojeve. Kako bi poduzeća ostala konkurentna, započela su različita mjerenja i praćenja mjernih rezultata i produktivnosti. Održavanje, koje je do tada bila aktivnost koja se poduzimala samo kada je do kvara došlo, postala je mnogo važnija za poduzeća te joj se danas pridodaje velika važnost. Sve ovo rezultiralo je razvijanjem novih strategija održavanja.

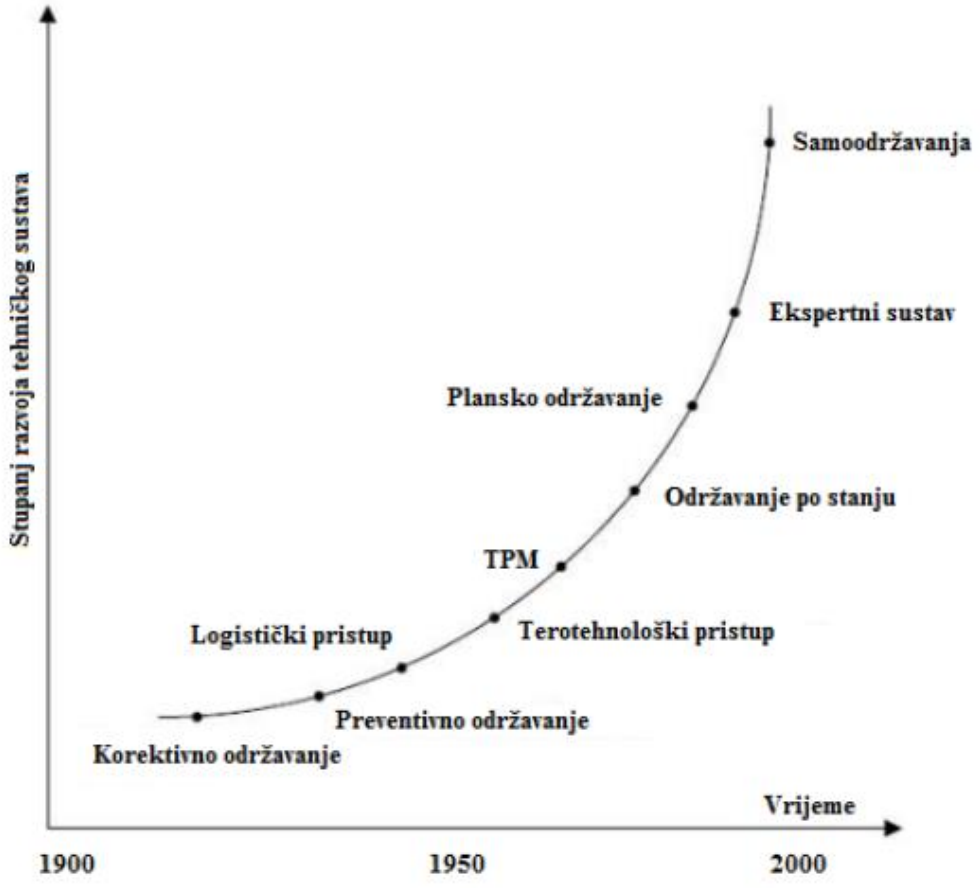
Troškovi održavanja predstavljaju glavni dio ukupnih operativnih troškova proizvodnje odnosno pogona. Zavisno o industriji, troškovi održavanja mogu se kretati između 15 i 60 posto troškova proizvodnje. Primjerice u prehrambenoj industriji, prosječni troškovi održavanja iznose oko 15 posto troškova proizvodnje, dok troškovi održavanja u proizvodnji željeza, čelika, papira i u teškoj industriji mogu dosezati i 60 posto ukupnih troškova. [4]

Prema istraživanjima o upravljanju održavanjem, pokazano je kako na jedan dolar, trećina tog dolara odlazi upravo na troškove izazvane nepotrebnim ili nepravilnim održavanjem [4]. Činjenica da se u SAD-u godišnje troši oko 200 milijardi dolara svake godine na održavanje [4], daje do znanja koliko zapravo održavanje ima veliki utjecaj te može biti jasno da su produktivnost i profitabilnost usko povezane s održavanjem. Uz ove podatke, činjenica je da se lošim održavanjem utječe na kvalitetu proizvedenih dijelova kojima poduzeća konkuriraju na tržištu.

Jedan od glavnih razloga neadekvatnog upravljanja održavanjem jest nedostatak činjeničnih podataka kojima će se kvantificirati stvarna potreba za popravkom ili održavanjem postrojenja, opreme ili sustava [4]. Naglim razvojem mikroprocesorske industrije te povezivanje instrumenata za mjerenje i prikupljanje podataka sa računalnim sustavima kojima se omogućuje njihova obrada, dolazi se do prilike za omogućavanjem kvalitetnijeg upravljanja i izvođenja operacija održavanja strojeva i opreme.

Pod strategijama održavanja podrazumijeva se skup tehnika i metoda sa ciljem smanjenja ili postizanja potpune eliminacije zastoja, uz minimalne troškove održavanja. Strategije održavanja mogu se podijeliti na reaktivne i proaktivne [5]. Pod reaktivne strategije održavanja spada Korektivno održavanje, dok pod proaktivno održavanje spada Preventivno održavanje.

Razvojem tehničkih sustava početkom 20. stoljeća potaknuto je i traženje rješenja kako bi tehnički sustavi bili u funkciji što dulje. Početkom 20-og stoljeća tehnička oprema postaje sve složenija, pojavljuju se novi materijali i konstrukcijska rješenja. To je pogodovalo da se u svijetu razvije niz koncepcija i modela održavanja odnosno suvremenih strategija održavanja [6]. Pod suvremenim strategijama održavanja mogu se smatrati*: Logistički pristup, Terotehnološki pristup, Održavanje po stanju, Plansko održavanje, Ekspertni sustavi, Model samoodržavanja.* Razvoj strategija održavanja vidljiv je na slici 1.



Slika 1. Razvoj koncepcija održavanja [6]

2.1. Korektivno održavanje

Korektivno održavanje (engl. Corrective Maintenance -On Failure Maintenance) predstavlja najstariji oblik održavanja. Filozofija ovog održavanja može se jednostavno opisati kao: Ako nije pokvareno, ne diraj. Ukoliko se pokvari, popravi. Na prvi pogled, ovakav stil održavanja čini se logičnim jer se ne troše nikakva sredstva na održavanje dok je oprema u stanju rada, a trošak nastaje tek kada se stroj pokvari. Međutim, ova tehnika održavanja predstavlja najskuplju tehniku održavanja. Razlog tome je što onog trena kada pogon stane, staje i proizvodnja dok se ne otkloni kvar, odnosno poduzeće ima nisku proizvodnu dostupnost. Dodatan trošak predstavlja i potreba za većim brojem osoblja zaduženih za održavanje kako bi otklonili kvarove, potreba za zalihama rezervnih dijelova te nepredvidivost kvarova i nemogućnost planiranja aktivnosti održavanja. Ovom strategijom održavanja, sustav traje onoliko koliko i njegova najslabija karika. Oštećenje podsustava često ima za posljedicu oštećenje drugih podsustava. Ovaj pristup održavanju može se smatrati i kao upravljanje održavanjem bez održavanja.

Kako bi se minimizirao utjecaj neočekivanih kvarova, osoblje održavanja mora reagirati na sve kvarove strojeva. Analize pokazuju kako se za popravke provedene korektivnom metodom održavanja mjesečno troši i do tri puta više nego kod redovnih i preventivnih održavanja [4].

Za korektivno održavanje općenito se može reći da se sastoji od niza zahvata kojima se nakon kvara sustav vraća u radno stanje. Danas, ovaj pristup se upotrebljava uglavnom za slučajeve kada kvar sustava ne može dovesti do [7]:

- Ugrožavanja sigurnosti,
- Težih havarija i lomova,
- Dugih zastoja,
- Velikih troškova održavanja,
- Neučinkovitog iskorištenja osoblja održavanja.

2.2. Preventivno održavanje

Ova strategija održavanja početak primjene doživjela je u SAD-u nakon drugog svjetskog rata, a brzo nakon toga se proširila po cijelom svijetu te je postala strategija na čijoj bazi su razvijene mnoge nove strategije [3]. Preventivnom strategijom održavanje se provodi i prije nego nastupi kvar i to na taj način su uklonjene sve negativne značajke korektivnog održavanja. Preventivno održavanje sastoji se od sistematskih kontinuiranih planskih pregleda, podmazivanja i popravaka strojeva, uređaja i zgrada, kako bi se održali u optimalnim uvjetima [6]. Ovom metodom nastoji se otkriti početak degradacije sustava prije nego nastane značajno oštećenje sustava, a izvršava se na nekoliko načina. Operacije ovog održavanja planirane su na statističkim značajkama dosadašnjih kvarova dijelova i strojeva.

Planiranjem održavanja se mogu umanjiti troškovi te se oprema može održavati u vremenu kada nije potrebna odnosno nije u radu pa se na taj način smanjuje vrijeme popravka. Istraživanja pokazuju da se može uštedjeti od 12% do 18% u odnosu na reaktivna održavanja [8]. Također, zbog manje učestalosti pojave kvara sigurnost je na višoj razini. Neki nedostaci preventivnog održavanja naspram korektivnog održavanja je primjerice zahtijevanje za planiranjem koje traži ulaganje vremena i sredstava. Učestalost preventivnog održavanja je često previše visoka, što prouzrokuje troškove, a uvođenjem praćenja stanja i analiziranjem se može smanjiti učestalost preventivnog održavanja bez smanjenja pouzdanosti. Preventivnim održavanjem obuhvaćene su sljedeće aktivnosti [7]:

- Preventivni periodični pregledi, čišćenja i podmazivanja,
- Traženje i otklanjanje slabih mjesta,
- Kontrolni pregledi,
- Planiranje popravaka.

Preventivni periodični pregledi, čišćenja i podmazivanja su unaprijed planirane i pripremljene aktivnosti koje se provode radi pravovremenog otkrivanja vremenskih kvarova. Njima se obuhvaća sva vrste uvida u stanje sustava, ali bez demontaže. Kontrolni pregledi su slični preventivnim pregledima, međutim kontrolni pregledi zahtijevaju poseban alat i instrumente za mjerenje te djelatnike osposobljene za takve preglede. Planski popravci su unaprijed pripremljeni radovi sa ciljem zamjene istrošenih dijelova [7].

Preventivno održavanje pogodno je za primjenu na opremi koja je kritična za rad, koja ima vjerojatnost povećanja kvarova s porastom korištenja, odnosno vremena u radu te opreme koja ima načine kvara koji mogu biti spriječeni redovitim održavanjem. Slučajevi u kojima

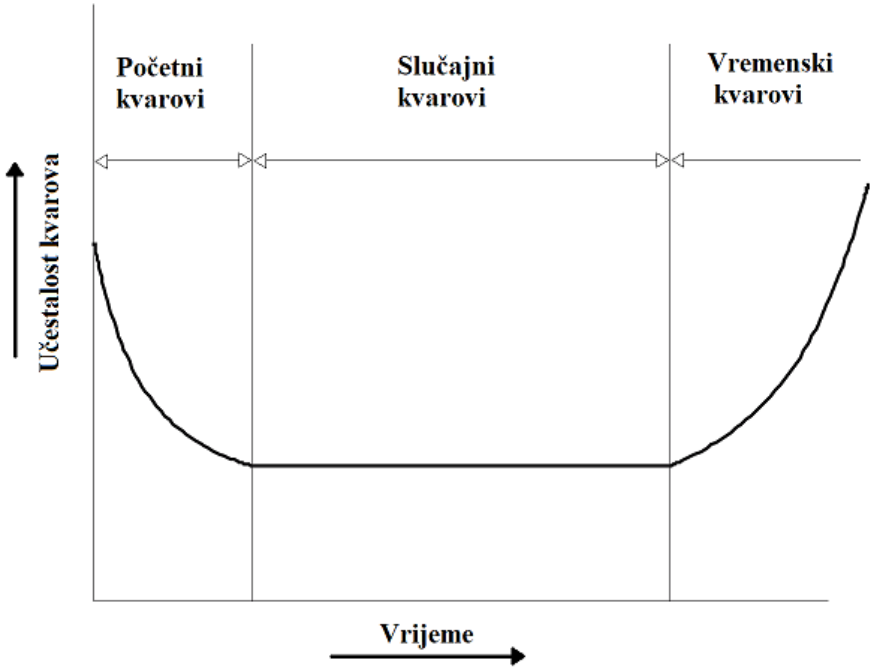
preventivno održavanje nije pogodno za primjenu su slučajevi kod kojih nastupaju slučajni kvarovi koji se ne mogu povezati s održavanjem te slučajevima koji nisu kritični za rad. [7]

Prednosti preventivne strategije održavanja [3]:

- Kvalitetnije upravljanje proizvodnim procesom,
- Osigurana željena kvaliteta proizvoda,
- Povećanje produktivnosti proizvodnog sustava,
- Veća iskoristivost raspoloživih resursa,
- Povećanje efikasnosti tehničkih sustava,
- Normiranje poslova održavanja,
- Smanjenje prekovremenog rada,
- Racionalno planiranje rezervnih dijelova,
- Povećanje sigurnosti i bolja kontrola zagađenja okoliša.

Nedostaci preventivne strategije održavanja [3]:

- Mogućnost pojave oštećenja,
- Visoka razina ranih kvarova,
- Neiskorišteni raspoloživi resursi,
- Visoki početni troškovi,
- Česti prekidi procesa eksploatacije sustava.



Slika 2. Krivulja "kade“ [5]

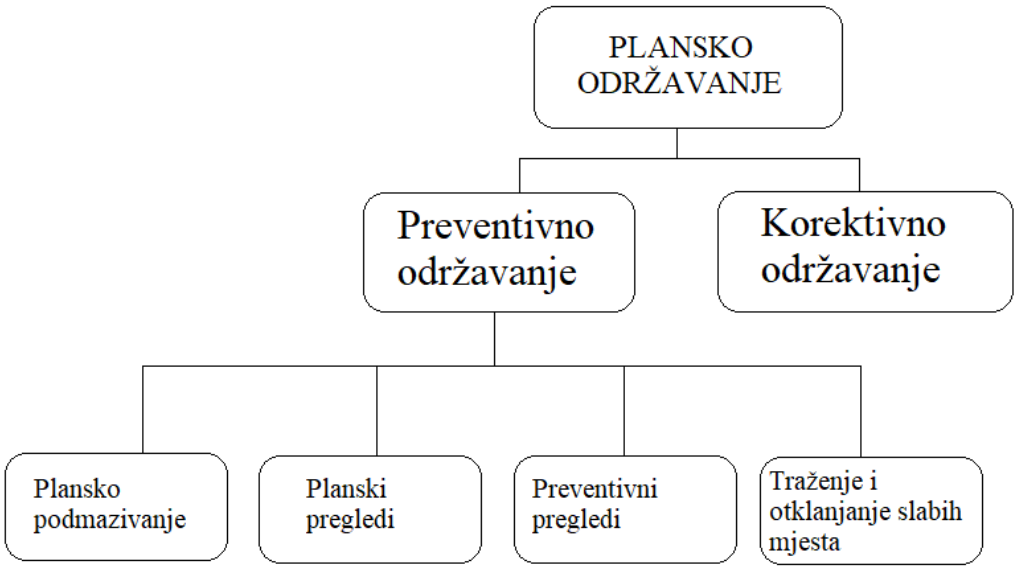
Sa slike 2. vidljiva je krivulja „kade“, odnosno dijagram učestalosti i vrsta kvarova. Slikom 2. se vidi da strojevi imaju visoku vjerojatnost kvara nakon instalacije, zatim dulji vijek trajanja sa niskom učestalosti kvarova te pred kraj životnog vijeka strojeva učestalost kvarova raste. U preventivnom održavanju, popravci su planirani pomoću statističkih krivulja, međutim glavni izazov ovakve strategije je što statističke krivulje nisu jednake za sve dijelove opreme u svim uvjetima rada.

2.3. Suvremeni koncepti održavanja

Razvoj složenosti tehničkih sustava te želja poduzeća da ostanu konkurentni, natjerala je na daljnji razvoj tradicionalnih strategija održavanja. Suvremenim strategijama održavanja nastoji se izvršiti planiranje održavanja unaprijed i reagiranje prije negoli do pojave kvara dođe kako bi efikasnost sustava ostala visoka.

2.3.1. Plansko održavanje

Plansko održavanje predstavlja kombinaciju korektivnog i preventivnog održavanja u omjeru koji najbolje odgovara određenom poduzeću [6]. Ono je planirano, određeno rasporedom i dokumentirano u svrhu provođenja održavanja prije nastanka kvara. Procesom planiranja održavanja povećava se učinkovitost zadataka održavanja i eliminiraju se utjecaji održavanja na operativnost sustava [7]. Operacije planskog održavanja sastoje se od niza aktivnosti koje se provode na određenom dijelu opreme ili sustava. Popravak mora biti unaprijed određen te ga izvodi kompletno osoblje zaduženo za održavanje kako bi se utvrdio ispravan rad opreme, odnosno eliminiralo vrijeme izgubljeno u kvaru. Planiranje se provodi prema proizvođačevim preporukama, za svaki predmet posebno. Ono mora biti zasnovano na podacima, temeljeno na radnom broju sati opreme ili nekom drugom parametru, a krajnji cilj je planskog održavanja je optimizacija korektivnog i preventivnog održavanja kako je prikazano na slici 3.



Slika 3. Plansko održavanje [7]

2.3.2. Logistički koncept održavanja

Ovaj pristup održavanju nastao je u SAD-u 1970-ih godina otprilike u isto vrijeme kada je nastala i terotehnološka koncepcija u Europi. Logistika je tehnička disciplina koja izučava rad, funkcioniranje i uvjete rada tehničkog sustava, obuhvaćajući sve osnovne aktivnosti njegova životnog ciklusa. Zadatak logistike je osiguranje svih čimbenika važnih za pouzdan rad sustava, za njegovo vraćanje iz stanja kvara u stanje rada. U slučaju održavanja radi se o logističkoj potpori proizvodnji, tj. tehnički sustav koji se promatraju od prvih ideja pri projektiranju, u samoj izgradnji i na kraju u eksploataciji. Težište je da se kroz projektiranje i proizvodnju opreme učini maksimalno kako bi oprema imala visok stupanj pouzdanosti i lako se održavala u svrhu učinkovite eksploatacije. Sa gledišta održavanja tehničkog sustava, logistika je inženjerski pristup problemima njegova održavanja od idejnog nastajanja do konačnog otpisa, s interakcijom svih čimbenika koji se javljaju u tome vremenskom periodu. Logistički sustav održavanja prije svega treba promatrati kroz pouzdanost, raspoloživost i pogodnost za održavanje, što su važne značajke tehničkog sustava. Ekonomski razlozi i suvremeni tehnički razvoj dovode do novih konfiguracija sustava. [7]

2.3.3. Terotehnološki pristup održavanju

Cilj svake dobro organizirane i upravljane funkcije održavanja poslovnih sustava je zadržavanje niske razine troškova koje čini zbroj troškova radnika na održavanju, troškova materijala i rezervnih dijelova te troškova koji nastaju kao proizvodni gubici, zbog zaustavljanja proizvodnog procesa ili smanjenog obujma istog, uslijed nastalih kvarova ili redovitih remonta. Terotehnologija je znanstvena disciplina koja istražuje metode i zakonitosti menadžmenta trajnih materijalnih sredstava ili tehničkih poslovnih sredstava tijekom njihovog životnog vijeka trajanja. Do razvoja terotehnologije došlo je 1970-ih godina u Europi. Prvu definiciju terotehnologije, koja je do danas gotovo ostala neizmijenjena postavio je Odbor Ministarstva za industriju u Velikoj Britaniji. Definicija glasi: "Kombinacija menadžmenta, financiranja, inženjeringa, izgradnje i drugih disciplina primijenjenih na fizička poslovna sredstva poduzeća s aspekta ekonomičnosti troškova tijekom njihova korisnog vijeka trajanja." Ovu definiciju najčešće prate i dvije napomene:

- Terotehnologija se bavi specifikacijom (potankim opisivanjem) i dizajniranjem za pouzdanost i održivost fizičkih poslovnih sustava poduzeća kao što su tvornice, strojevi, oprema, zgrade i postrojenja. Primjena terotehnologije uzima u obzir i cjelokupni proces instalacije, komisije, upotrebe, održavanja, modificiranja i zamjene sustava. Odluke su

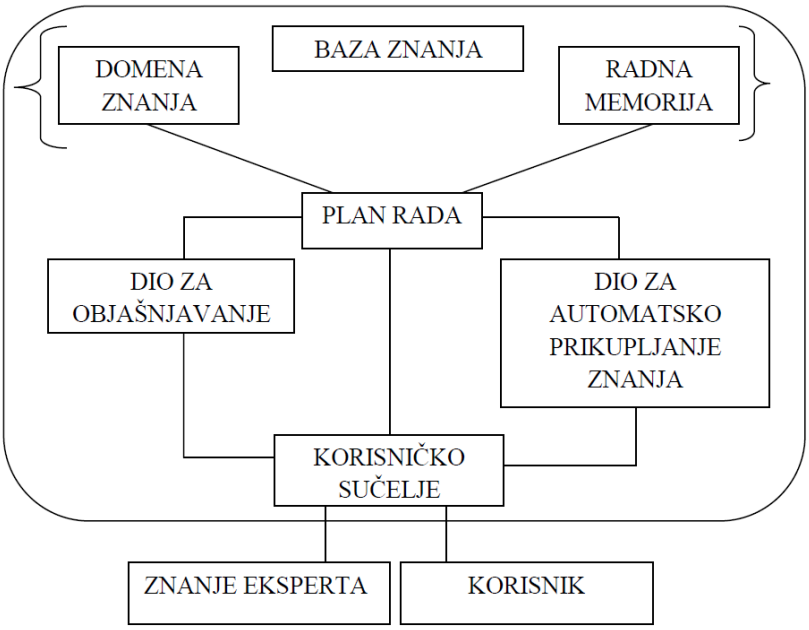
uvjetovane povratnim informacijama o dizajnu, performansama i troškovima tijekom cijelog životnog vijeka projekta.

- Terotehnologija se podjednako odnosi i na sredstva i na rezultate proizvodnje jer proizvod jednog poduzeća često postaje sredstvo drugog poduzeća. Čak i kada je proizvod jednostavan predmet potrošnje, terotehnologija može pozitivno utjecati na njegov dizajn i privlačnost za kupca, a to će se reflektirati u povećanoj tržišnoj sigurnosti proizvođača.

Najjednostavnije rečeno, konceptom terotehnologije promovira se primjena svih potrebnih tehnika kojima se može osigurati da korisnik sustava dobije najveću vrijednost za svoj novac. [9]

2.3.4. Ekspertni sustavi

Ekspertni sustavi održavanja javljaju se 1980-ih godina i to zahvaljujući razvojem opreme i programske podrške. Ekspertni sustavi su računalni programi realizirani različitim metodama umjetne inteligencije, koji poput ljudi rješavaju probleme u svom specijaliziranom problemskom području uz pomoć logičkog zaključivanja i upotrebom opsežnog znanja koje je u njih implementirao čovjek [10]. Danas se pod tim pojmom podrazumijeva sustav koji može učinkovito obavljati funkcije upravljanja, i to u uvjetima u kojima postoje nepotpune i nepouzdane informacije, smetnje itd. Ovakvi sustavi svojom inteligencijom imaju sposobnost učenja i prilagođavanja te zaključivanja i donošenja odluka. Struktura tipičnog ekspertnog sustava prikazana je na slici 4.



Slika 4. Elementi ekspertnog sustava [7]

Sa slike 4. vidljive su komponente ekspertnog sustava a to su [7]:

- Korisničko sučelje - mehanizam putem kojeg se odvija komunikacija između korisnika i ekspertnog sustava.
- Dio za objašnjenje – služi korisniku za pojašnjenje principa rasuđivanja sustava.
- Dio za zaključivanje – modul kod ekspertnih sustava koji generira algoritme za rješavanje problema koristeći bazu znanja za problemsko područje naziva se mehanizmom zaključivanja.
- Plan rada – predstavlja listu pravila nastalih putem procesa zaključivanja, čiji su obrasci ispunjeni preko podataka i ciljeva u radnoj memoriji.
- Sredstvo za automatsko prikupljanje znanja – automatiziran način za korisnika da dođe do znanja sustava, predstavlja pristupačniji pristup nego da je znanje kodirano.
- Baza znanja – Kvaliteta donošenja odluka ekspertnog sustava ovisi o obujmu i kvaliteti znanja koje ono posjeduje.

Ekspertni sustavi moraju moći primati i obrađivati informacije tijekom čitavog vremena rada, razvrstavati te informacije i uspoređivati ih sa podacima unutar baze podataka kako bi mogao prepoznati potencijalni problem. Pravovremenost primjene znanja iz baze podataka bitan je faktor, kako bi korisnik ekspertnih sustava pravovremeno mogao otkloniti potencijali kvar. Također moraju moći, kao što je već navedeno, primati i unaprjeđivati svoja prvobitna znanja te ih moraju moći primjenjivati na daljnje slučajeve. Ekspertni sustavi se zapravo ponašaju

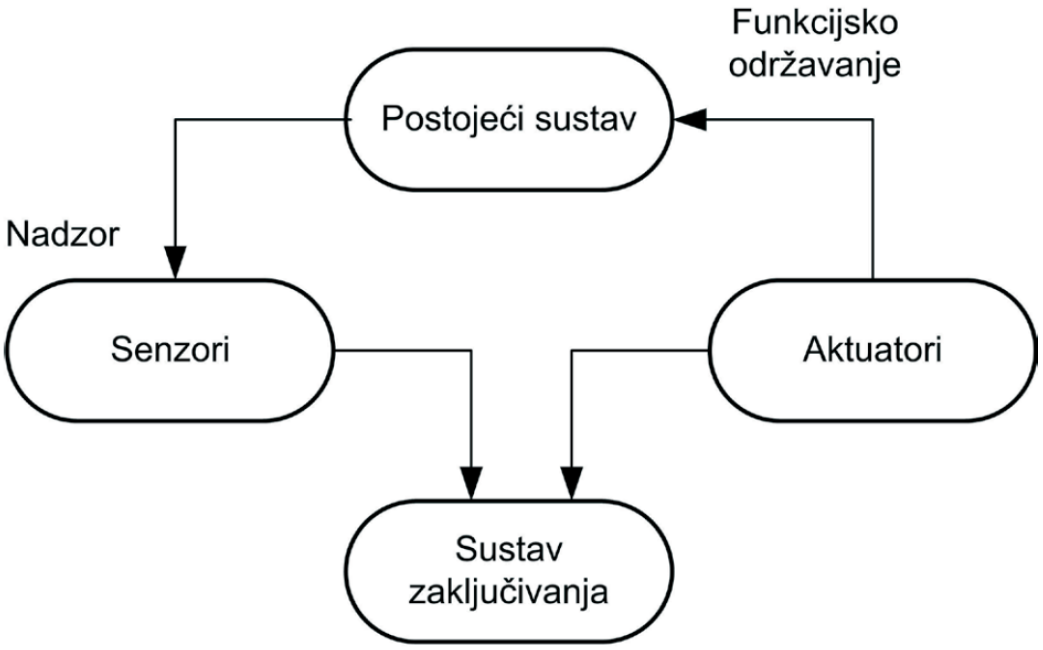
poput stručnjaka za određeno područje, koji daju rješenje na problem uz objašnjenje svih svojih odluka. Odlučivanje računala temelji se na izboru ponuđenih unosa u bazi, vezanim uz određene primljene podatke te se sastoji od sljedećih koraka. U prvom koraku, nakon što korisnik unese podatke u računalo, ono prepoznaje problem, određuje moguće uzroke problema te predlaže rješenje korisniku uz obrazloženja kako je došlo do tih rješenja. U idućem koraku korisnik mora ili prihvatiti rješenje ili odbiti rješenje. U slučaju odbijanja rješenja, računalo se vraća na korak određivanja uzroka problema i predlaganja novog rješenja, sve dokle god korisnik ne prihvati rješenje. Nakon što korisnik prihvati rješenje, on ga primjenjuje te daje ocjenu dobivenog rješenja računalu. Ukoliko je rješenje neučinkovito, računalo ponovno određuje novi uzrok te predlaže novo rješenje. U slučajevima da računalo daje netočna rješenja odnosno dijagnozu u području održavanja, tada je potrebno nadopuniti sustav novim saznanjima, jer baza podataka računala ne pohranjuje apsolutno sve moguće slučajeve koji se mogu pojaviti tijekom rada stroja.

2.3.5. Samoodržavanje

Kako bi se spriječio otkaz zbog pojave kvara kod tehničkih sustava, proizvođači takve uređaje nastoje dizajnirati ne da su samo tolerantni na pojavu greške, već i da tu grešku mogu otkloniti. Pod ovime se ne podrazumijeva popravak ili zamjena fizičkih dijelova, već na popravak funkcionalnosti sustava što znači da se u slučaju greške sustav se sam mora vratiti u ispravno funkcionalno stanje. Primjenom samoodržavanja dolazi do povećanja raspoloživosti sustava, a i poboljšavanja tolerantnosti na greške. Pretpostavka za implementiranje sustava samodržavanja na nekom sustavu je da je na njemu ugrađen sustav nadzora i samodijagnostike. Sustav samoodržavanja mora imati sposobnost izvođenja :

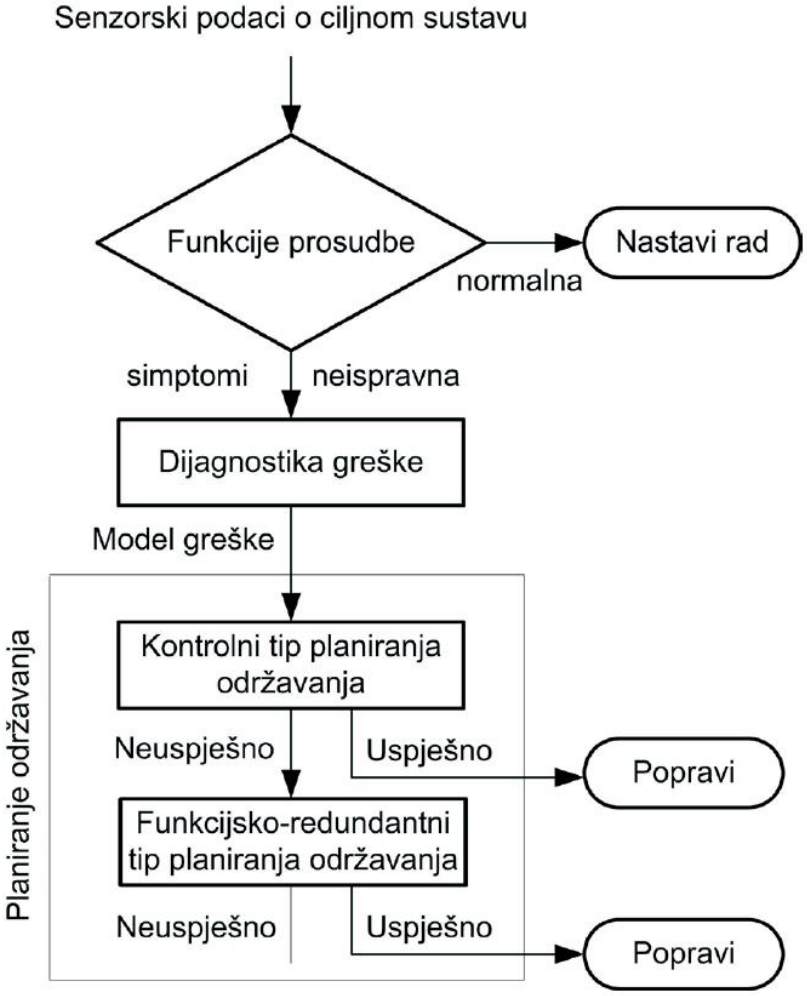
- Nadzora,
- Procjene greške,
- Dijagnosticiranja,
- Planiranja održavanja
- Održavanja.

Na slici 5. vidi se arhitektura sustava samoodržavanja:



Slika 5. Arhitektura sustava samoodržavanja [11]

Sustavi samoodržavanja imaju nekoliko nedostataka. Prvi nedostatak je da sustav može obaviti samo one operacije održavanja koje ne zahtijevaju pričuvne dijelove. Time se ograničava vrsta grešaka koju sustav može popraviti. Drugi nedostatak je da ovakvo održavanje predstavlja samo privremenu mjeru održavanja, čime se samo odgađa fizičko održavanje. Uz navedene nedostatke održavanja, sustav samoodržavanja uključuje dva pristupa održavanju: kontrolni i funkcijsko- redundantni. Kontrolnim pristupom se izvršava samopodešavanje radnog stanja bez mijenjanja ili reorganizacije sustava. Funkcijsko-redundantni pristup ima mogućnost restrukturiranje ponašanja kako bi se održalo obavljanje zahtijevane funkcije, međutim to rezultira i neželjenim troškovima te utječe na fizičke karakteristike sustava. Na slici 6. prikazan je algoritam sustava zaključivanja.



Slika 6. Algoritam zaključivanja kod sustava samoodržavanja [11]

2.4. Održavanje po stanju (prediktivno održavanje)

Održavanje po stanju oblik je preventivnih aktivnosti jer se izvodi prije nastanka kvara, ali je inicirano kao rezultat poznavanja stanja postrojenja ili njegovih komponenata odnosno stanja koje je poznato kroz određeni vid kontrole [12]. Nad sustavom se provodi neprekidna kontrola uz pomoć senzora pomoću koje se prati stanje sustava. Analizirajući informacije sa senzora donose se odluke o roku i opsegu aktivnosti održavanja. Iz iskustava u eksploataciji, pokazano je da većinom tehnički sustavi ne gube svoju funkcionalnost odjednom, već postupno. Mjerenjem parametara bitnih za ocjenjivanje stanja tehničkog sustava može se zaključiti funkcionira li promatrani sustav unutar granica normalnog rada ili se javljaju odstupanja u radu tog sustava te se na temelju tih podataka mogu se donijeti pravovremene odluke. Razlika između preventivnog održavanja i održavanja po stanju je u to me što se pri preventivnom održavanju zahvat održavanja provodi prema unaprijed utvrđenom vremenu, a održavanjem po stanju postupak održavanja provodi se u slučaju kada zadani sustav izađe iz granica normalnog rada [7]. Zadatak ove strategije je detekcija ranih stadija kvarova opreme te iniciranje akcije održavanja u pravo vrijeme kako bi se potreban rad za popravak mogao odraditi prije nastanka kvara. U industrijski integriranom obliku, strategija prediktivnog održavanja podrazumijeva [13]:

- 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.

Prednosti održavanja po stanju su [8]:

- Povećanje operativnog životnog vijeka komponenti,
- Ušteda energije,
- Veća kvaliteta proizvoda,
- Smanjenje cijene dijelova i troškova radne snage,
- Procjena od 8% do 12% uštede u odnosu na preventivno održavanje,
- Smanjenje vremena nedostupnosti opreme.

Nedostaci održavanja po stanju [8]:

- Povećani trošak investicija u dijagnostičku opremu,
- Povećani troškovi investicija u obuku radnika,
- Potencijalne uštede nisu jasno vidljive menadžmentu.

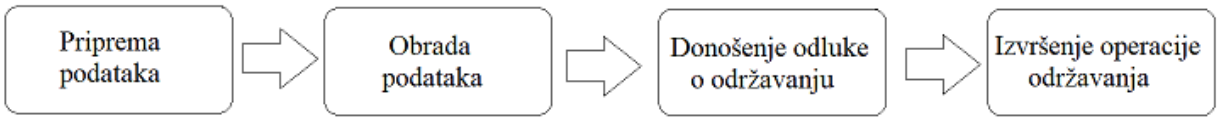
Prema [8] pokazuje se da se investicije s vremenom isplate:

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

U održavanju po stanju često se spominje termin dijagnostika. Dijagnostika u održavanju treba ustvrditi stanje sustava ili dijela sustava bez njegovog demontiranja, a poželjno i bez njegovog zaustavljanja [12]. Dijagnostikom se omogućava procjena stanja opreme i donošenje odluke o potrebnim aktivnostima održavanja, na temelju izvršenih mjerenja unaprijed odabranog parametra ili odabranih parametara. Prednost dijagnostike je što se odluke donose uz pomoć rezultata mjerenja bazirane na stvarnim podacima koji opisuju stanje promatrane opreme. Neke od tehnika za praćenje stanja su [13]:

- Mjerenje i analiza vibracija,
- Termografija,
- Ultrazvuk,
- Akustična emisija,
- Analiza maziva.

Održavanje po stanju sastoji se od sljedećih koraka: Prikupljanje podataka, Obrada podataka, Donošenje odluke, Izvršavanje odluke (slika 7.) [13].



Slika 7. Koncept održavanja po stanju [13]

Prikupljanje podataka ostvaruje se raznim senzorima kojima se promatraju željene komponente te se pohranjuju njihove vrijednosti tokom vremena. Sami podaci prikupljeni iz senzora nemaju preveliko značenje, stoga se provodi njihova obrada kako bi se podaci mogli interpretirati i omogućiti donošenje daljnjih odluka.

Dijagnostika se uglavnom fokusira na pronalaženje i izolaciju greške u dijelovima sustava. Međutim, u novije doba, mogućnost detekcije i izolacije grešaka i njihovo predviđanje na temelju dostupnih dijagnostičkih podataka iz prošlosti postaje sve veći prioritet istraživačkih tema [14]. Glavna razlika između dijagnostike i prognostike je činjenica da dijagnostika predstavlja analizu sustava nakon događanja, a prognostika prije događanja. Prema [15] procjenjuje se da održavanje orijentirano prognostici može smanjiti troškove i do 25%. Prognostikom se podrazumijeva primjena automatiziranih metoda detekcije, dijagnostike i analize degradacije fizičkog sustava, uz izračun preostalog iskoristivog vijeka sustava do pojave kvara.

Održavanje po stanju, koje se sastoji od dijagnosticiranja kvarova i prognoziranja budućih kvarova gotovo se ne može zamisliti bez strojnog učenja (poglavlje 3.). Senzorima se prikupljaju podaci koji se obrađuju uz pomoć računala, a razvojem strojnog učenja, to računalo može identificirati određene uzorke ponašanja u podacima, naučiti ih te na temelju toga donositi određene zaključke. Imajući dovoljno veliku bazu podataka prikupljenu senzorima, računalo preko naučenih obrazaca može predviđati ponašanje opreme te predviđati kraj njihovog životnog vijeka.

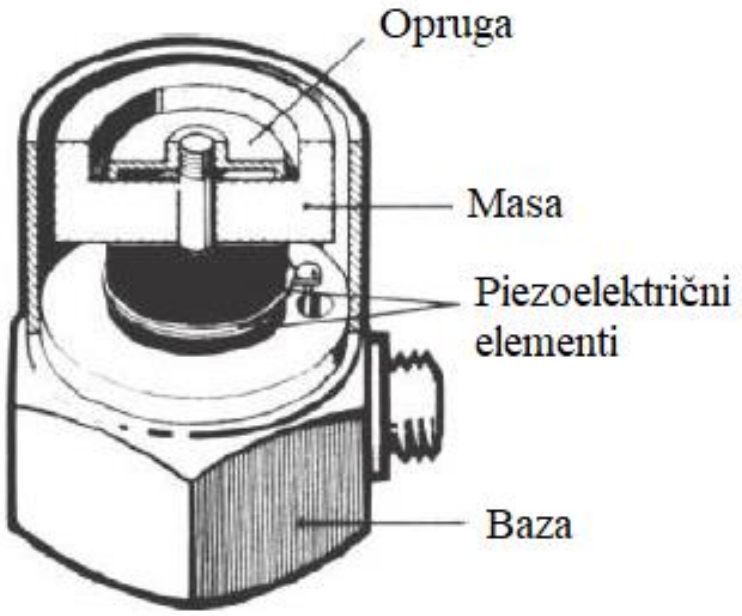
2.5. Mjerenje i analiza vibracija

Tema ovog rada je procjena tipa uzroka kvara i intenziteta kvara kotrljajućeg ležaja te će se podaci potrebni za obradu temeljiti na podacima dobivenim metodama mjerenja i analize vibracija.

Strojevi proizvode vibracije koje su uglavnom povezane sa periodičkim operacijama strojeva kao što su rotacije. Frekvencije na kojima se takvi događaji ponavljaju često daju direktnu indikaciju izvora te vibracije, stoga se primjenjuju razni alati za analizu vibracija [16]. Analiza vibracija predstavlja najsnažniju i najčešće korištenu tehniku kod održavanja po stanju za strojne dijelove unutar industrijskih sustava. Amplituda svake pojedine vibracijske komponente ostat će konstanta ukoliko dinamika tehničkog sustava ostane nepromijenjena. Stoga, prvi znaci kvarova na opremi vibracijskom analizom mogu se uočiti prije pojave kvara. Također, svaki tip kvara ima karakterističnu komponentu vibracijskih frekvencija koje se mogu filtrirati i definirati. Vibracijsku analizu moguće je provoditi konstantno, a rezultate obrade moguće je

snimati u približno realnom vremenu [13]. Strojni dijelovi mogu imati trajno pričvršćene elektroničke pretvarače signala (senzore) koji konstantno prate rad stroja te u slučaju naglih promjena vibracija mogu biti ugašeni u kratkom trenutku, sprječavajući daljnje kvarove koji bi mogli dovesti do katastrofalnih kvarova. Vibracijska analiza pogodna je kod uočavanja raznih vrsta kvarova kao što su neravnoteža, ekscentričnost, pukotine, istrošeni dijelovi, iskrivljenost itd.

Kao alati za mjerenje vibracija primjenjuju se senzori akcelerometri. Akcelerometri su pretvarači koji daju signal proporcionalan akceleraciji. Najčešće primjenjivani akcelerometri koji se primjenjuju za praćenje stanja strojeva su piezoelektrični akcelerometri koji koriste piezoelektrična svojstva određenih kristala i keramike [16]. Na slici 8. se može vidjeti dizajn kompresijskog piezoelektričnog akcelerometra.



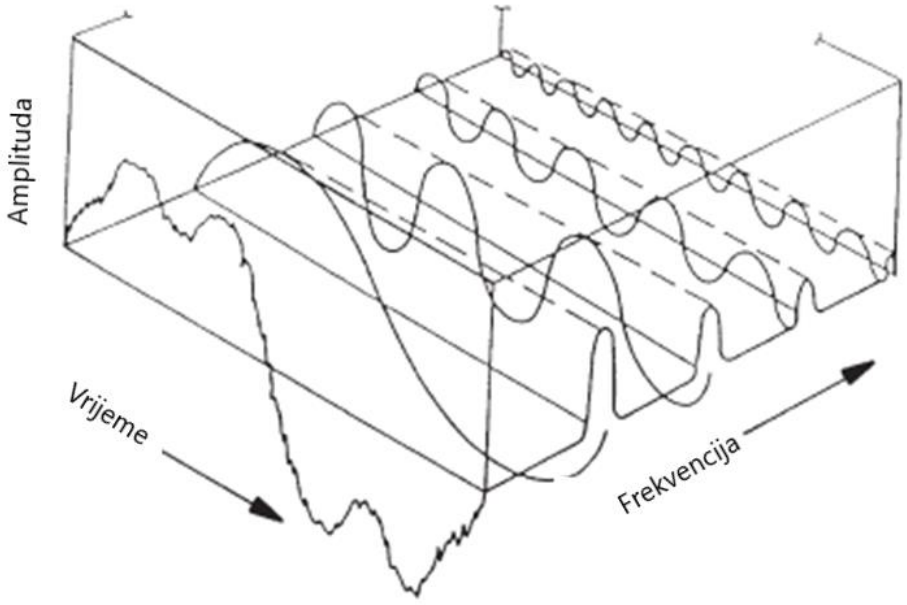
Slika 8. Elementi kompresijskog piezoelektičnog akcelerometra [16]

Kada je baza akcelerometra postavljena na vibrirajući objekt, masa je prisiljena pratiti kretanje baze preko piezoelektričnog elementa koji se ponaša kao kruta opruga. Varirajuća inercijska sila mase uzrokuje blagu deformaciju piezoelektričnih elemenata, stvarajući naprezanje koje je proporcionalno promjeni akceleracije. Time se proizvodi električni signal koji je proporcionalan akceleraciji, a mjeri se u $\frac{pC}{ms^2}$ (pikokuloni po metru sekundi na kvadrat). Ovaj signal potrebno je pretvoriti u napon i pojačalom ga pojačati do potrebne mjere [27].

Vibracijski profili mogu biti prikupljeni i prikazani u dva različita oblika [13]:

- Vremenska domena,
- Frekvencijska domena.

U vremenskoj domeni, vibracijski signali prikazani su kao amplitude u odnosu na vrijeme. Profil vibracija u vremenskoj domeni zbroj je svih vibracijskih komponenti koje su prisutne na opremi tijekom mjerenja. Samim time, teško je izravno vidjeti utjecaj određenih izvora vibracija. Stoga, u analizi vibracijskog signala primjenjuje se analiza u frekvencijskoj domeni gdje se vremenski trag signala rastavlja na diskretne frekvencijske komponente. Ovo se ostvaruje matematičkim procesom zvanim „Fourierova transformacija“. U frekvencijskoj domeni, na vertikalnoj osi je amplituda vibracija, a na horizontalnoj osi su frekvencije vibracija, dok su u vremenskoj domeni na horizontalnoj osi frekvencije, a vrijeme je na vertikalnoj osi. Prednost analize vibracija u frekvencijskoj domeni jest mogućnost normalizacije svake vibracijske komponente, tako da se složeni spektar stroja može razložiti na diskretne komponente [4]. Prema teoriji Fourierove serije, svaki periodički signal može se predstaviti sumom pravilnih sinus valova, a to vrijedi i za kompleksne vibracijske signale [13]. Primjenom spektralne analize, može se ustanoviti na kojim frekvencijama vibriranja se pojavljuju veće amplitude te se time može pronaći određena karakteristika kvarova opreme [13].
Na slici 9. prikazan je odnos između vremenske domene i frekvencijske domene [4].



Slika 9. Odnos vremenske i frekvencijske domene [4]

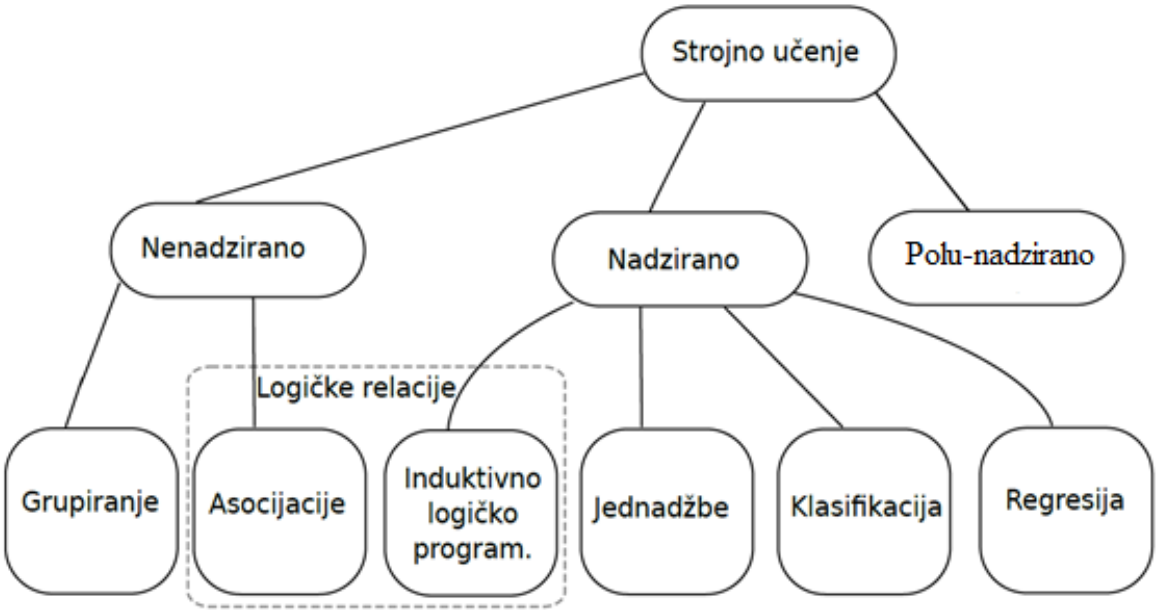
3. STROJNO UČENJE

Kao i kod održavanja, ne postoji samo jedna prihvaćena definicija za strojno učenje. Jedna od starijih definicija strojnog učenja je definicija Arthura Samuel-a iz 1959. godine: „To je znanost koja računalu daje mogućnost učenja bez izričitog programiranja“ [17]. Samuel je napisao računalni program za igru *Dama*, bez da je programirao strategije igre. Računalo je igralo oko desetak tisuća partija protiv Samuela i snimalo koje pozicije na igraćoj ploči dovode do kojih rezultata. Tokom vremena računalo je naučilo koje pozicije na igraćoj ploči dovode do pobjede te je porazilo Samuela.

Moderniju i formalniju definiciju strojnog učenja dao je Tom Michell 1998. godine: „Za računalni program se može reći da uči iz *iskustva E* za *zadatak T* sa mjerom *učinkovitosti P,* ako njegova *učinkovitost* na *T*, mjerena sa *P*, poboljšava *iskustvom E*“ [17]. Shodno Samuelovom primjerom, ova definicija se može interpretirati kao: računalni program koji uči igrati damu može poboljšati svoju učinkovitost mjereći sposobnost ostvarivanja pobjede, kroz iskustvo dobiveno igranjem protiv samoga sebe.

Analizom ovih definicija može se zaključiti da strojno učenje predstavlja kategoriju algoritama koja dopušta programima da budu točniji u predviđanju ishoda, bez da se izričito programiraju, temeljeći se na podacima iz prošlosti. Znajući ovo, može se zaključiti kako strojno učenje spada pod područje umjetne inteligencije koje se definira kao sposobnost snalaženja u nepoznatim situacijama (inteligencija) za nežive sustave.

Sustavi sa strojnim učenjem često se opisuju kao oni koji uče iz iskustva uz ljudski nadzor ili bez njega. Tako se može napraviti podjela na nadzirano strojno učenje (engl. *Supervised learning*) i na nenadzirano strojno učenje (engl. *Unsupervised learning*). U nadziranom strojnom učenju računalni program uči na temelju unaprijed pripremljenih podataka kojima su poznati ulazi i izlazi. Nenadziranim strojnim učenjem, računalni program ne uči iz označenog skupa podataka sa poznatim izlazima, već sam pokušava otkriti ponašanje u podacima. Postoje problemi koji primjenjuju i nadzirano i nenadzirano učenje te se na takve probleme primjenjuje strojno učenje imena polu-nadzirano učenje (engl. *Semi-Supervised Learning*) kod kojeg računalni program dobije odgovor za odluke koje je donio [18]. Prikaz raščlambe strojnog učenja prikazan je na slici 10.



Slika 10. Raščlamba Strojnog učenja [19]

Kod nadziranog i polu-nadziranog učenja, tijekom učenja podaci se mogu podijeliti na trening, test i validacijske skupove podataka. Trening skup podataka služi kao skup podataka na kojima algoritam uči, sa testnim podacima algoritam procjenjuje svoju efikasnost na novim podacima, odnosno podacima koji ne spadaju u trening set. Validacijski skup podataka može poslužiti kao nepristrani skup podataka za evaluaciju modela. Njime se pomaže u prilagođavanju hiperparametara i konstrukciji cijelog modela kako bi algoritam davao što točnije rezultate na testnim skupovima podataka.

3.1. Nadzirano učenje

Nadzirano učenje predstavlja najčešći oblik strojnog učenja. Za potrebu ovog učenja primjenjuju se označeni podaci, tj. vrijednosti ulaznih i izlaznih veličina. Funkcija koja povezuje ulazne i izlazne varijable nakon procesa učenja omogućuje predviđanje izlazne varijable za nove ulaze varijable u podacima koje želimo predvidjeti. Najjednostavniji oblik funkcije prema [20] je:

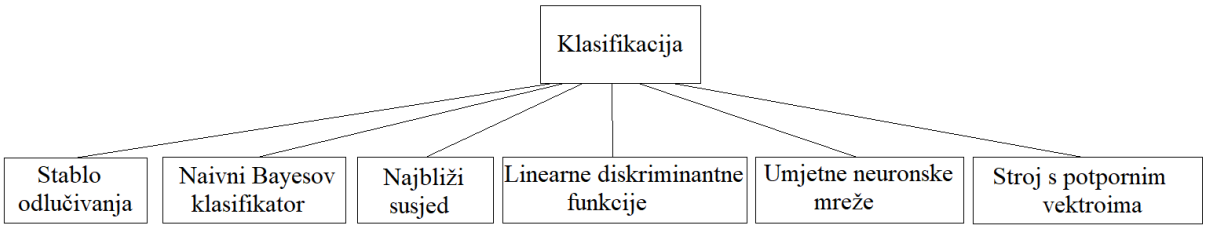
$$Y = f(X) \tag{1}$$

gdje X predstavlja ulazne varijable, Y izlaznu varijablu, a $f(X)$ predstavlja funkciju koja povezuje ulaz i izlaz. Dva najčešća problema koje algoritmi nadziranog učenja obrađuju su: klasifikacija i regresija. Kod klasifikacije, izlazna varijabla je diskretne vrijednosti, a kod regresijskog problema, izlazna varijabla je kontinuirane vrijednosti.

3.1.1. Klasifikacija

U ovoj vrsti problema nadziranog strojnog učenja, od računalnog programa se traži da odredi kojoj od k predefiniranih kategorija pripadaju ulazne vrijednosti. Kako bi se riješio ovaj problem, algoritam učenja mora naučiti funkciju $f: \mathbb{R}^n \rightarrow \{1, \ldots k\}$. Kada $y=f(x)$, model dodjeljuje ulaznu vrijednost x u kategoriju naučenih klasa k preko numeričke vrijednosti izlaza funkcije. [20]

Najčešći primjer korištenja ovog problema je svrstavanje elektroničke pošte kao željenu i neželjenu (engl. *spam*). Postoje razni algoritmi klasifikacije, a među najpoznatije spadaju: *stabla odlučivanja, pravila odlučivanja, naivni Bayesovi klasifikatori, Bayesove mreže vjerovanja, klasifikatori najbližih susjeda, linearne diskriminantne funkcije, logička regresija, strojevi s potpornim vektorima i umjetne neuronske mreže*. Na slici 11 prikazana je podjela klasifikatora.

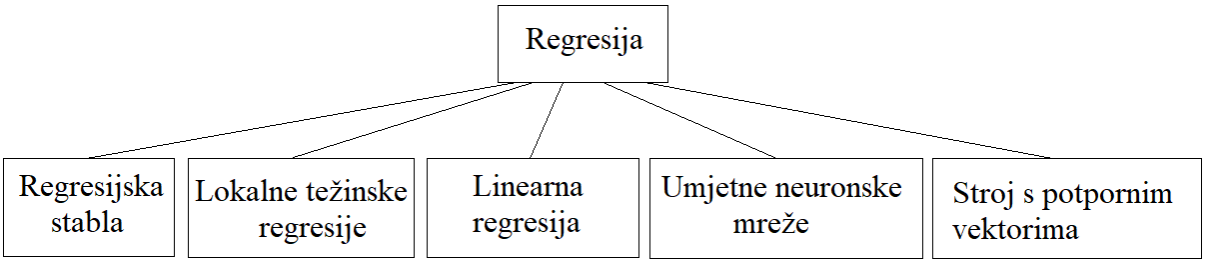


Slika 11. Klasifikacija [19]

3.1.2. Regresija

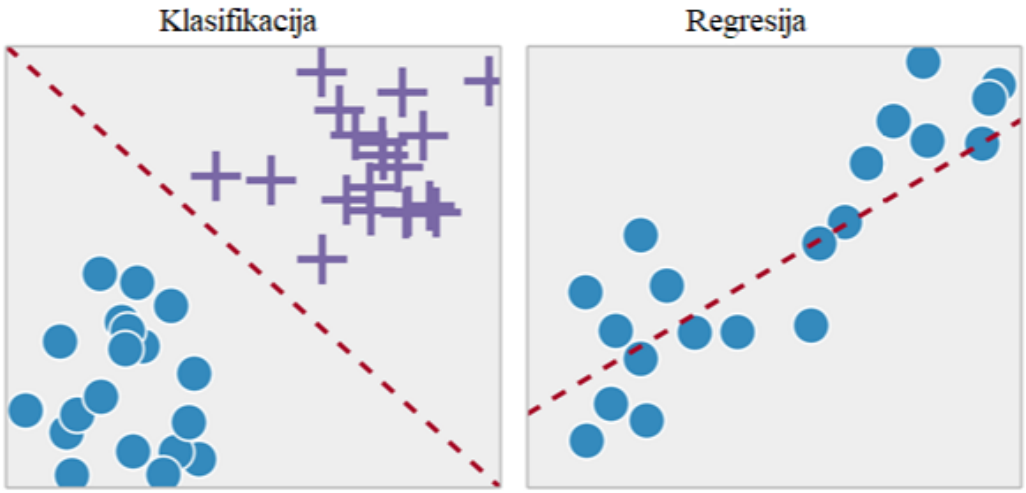
Kod problema regresije, zadatak računalnog programa jest predviđanje numeričke vrijednosti nakon zadavanja ulaznih varijabli. Kako bi se ovaj problem riješio, od algoritma učenja se traži da nauči funkciju $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Regresija je slična klasifikaciji, uz jedinu razliku da je vrsta izlazne vrijednosti drukčija. Kao najčešći primjer primjene regresije spada procjena predviđanja kretanja cijena na tržištu. [20]

Kao i klasifikacija, postoje više algoritama regresije: *linearne regresije, regresijskih stabala, lokalne težinske regresije, strojeva s potpornim vektorima za regresiju i višeslojnih neuronskih mreža za regresiju*. Podjela regresije dana je na slici 12.



Slika 12. Regresija [19]

Na slici 13 prikazana je razlika klasifikacije i regresije.



Slika 13. Razlika između klasifikacije i regresije [21]

3.1.3. Logičke relacije

Logičke relacije mogu se smatrati generaliziranim diskretnim funkcijama. Za razliku od klasifikacije, logičke relacije nemaju jedinstvenu ovisnu diskretnu varijablu, već se sve varijable promatraju jednako. U nekim slučajevima, varijable imaju nepoznate vrijednosti te ih se želi predvidjeti, a u drugim slučajevima poznate su sve vrijednosti, no samo se želi provjeriti da li relacije između određenih setova vrijednosti vrijedi. [19]

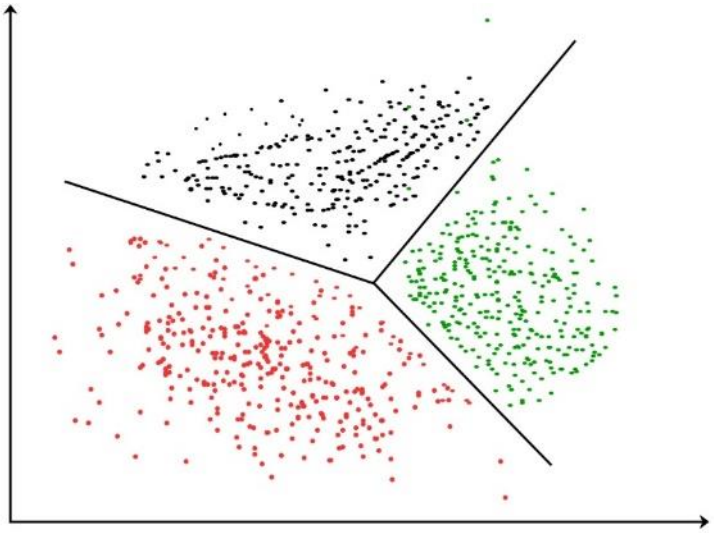
3.2. Nenadzirano učenje

Dok su kod naziranog učenja su odmah poznate izlazne vrijednosti koje bi se mogle povezati sa ulaznim vrijednostima, kod nenadziranog učenja one nisu poznate. Poznate su samo ulazne vrijednosti *(X)* i nema odgovarajućih izlaznih vrijednosti. Zadatak nenadziranog učenja jest modeliranje strukture podataka ili njihova distribucija kako bi se naučilo više o samim podacima. [22]

U jedan od najčešćih problema nenadziranog učenja spada problem grupiranja (engl. *clustering*).

Grupiranje

Za razliku od regresije i klasifikacije, kod grupiranja nema ciljne varijable, već su dane samo ulazne vrijednosti. Zadatak grupiranja je odrediti koherentne podskupine (klastere). Broj klastera može biti poznat unaprijed ili može biti određen algoritmom učenja. Najpoznatiji algoritmi grupiranja su: *grupiranje po k-srednjih vrijednosti (k-means), hijerarhijsko grupiranje i grupiranje po k najbližih susjeda (k-NN)* (slika 14.).



Slika 14. Grupiranje (k-means algoritam) [23]

3.3. Polu-nadzirano učenje

Problemi gdje velika količina podataka ima ulaznu vrijednost *(X)*, a samo neki od njih su određeni kao ciljna varijabla, odnosno izlazna vrijednost *(Y)*, spadaju u grupu polu-nadziranog učenja. Jedan od načina kako se može nositi sa ovim problemom je da se zanemare svi podaci koji nemaju izlaznu vrijednost *Y*, međutim onda se upravlja sa relativno malim skupom podataka. Ukoliko bi se željelo iskoristiti sav dostupan skup podataka kako bi se dobilo robusnije učenje, može se koristiti iskoristiti tehnika zvana lažno označavanje. Prvo se uzima

označeni skup podataka te se na njemu provode učenje, pomoću primjerice umjetne neuronske mreže. U idućem koraku nad neoznačenom skupu podataka provede se predviđanje pomoću umjetne neuronske mreže naučene na označenim podacima te im se dodjeljuje izlazna vrijednost (oznaka) koju odredi mreža. Nakon što su se označili svi podaci, ponovno se provede postupak učenja kao u prvom koraku, međutim ovaj put se koristi cijeli skup podataka (prvotno označeni i neoznačeni podaci). Iz ove tehnike vidljivo je da polu-nadzirano učenje zapravo predstavlja kombinaciju nadziranog i nenadziranog učenja**.**

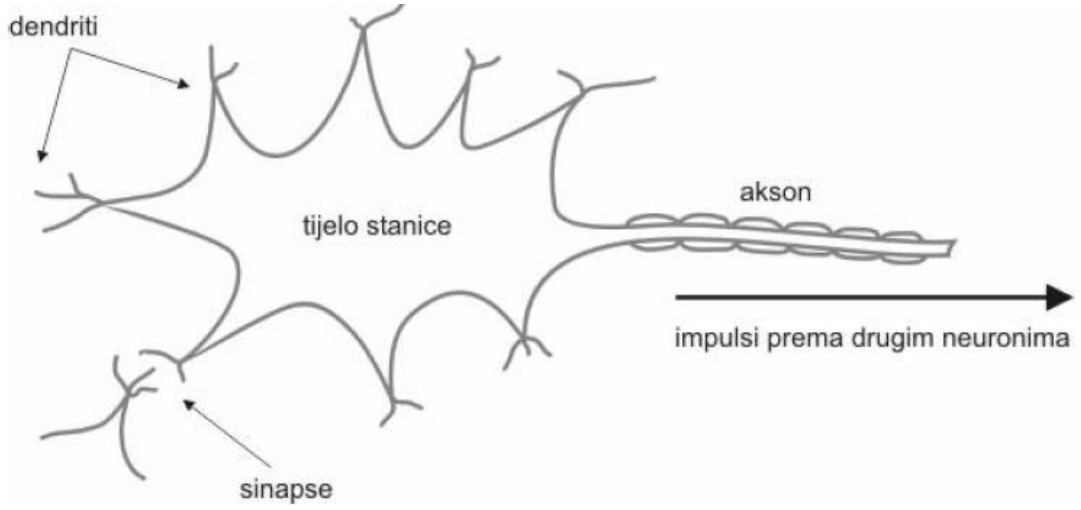
3.4. Duboko učenje

Duboko učenje spada u potkategoriju strojnog učenja, koje spada u područje umjetne inteligencije. Za strojno učenje, kako je navedeno u uvodnom dijelu poglavlja 3, rečeno je da je to proces učenja računala da izvrši određeni zadatak, bez da se izričito programira korak po korak za taj zadatak. Ono zahtjeva učenje te na kraju učenja daje precizna predviđanja kada se unesu ulazni podaci. Osnovni model strojnog učenja napreduje u izvršavanju svojih zadataka, no i dalje su im potrebne neke smjernice. U slučaju da strojno učenje kao izlaznu informaciju vraća netočne predikcije, potrebno je ručno podešavanje kako bi ostvarivalo točna predviđanja. Međutim, sa modelom dubokog učenja, algoritam sam utvrđuje radi li se o točnom ili netočnom predviđanju. U suštini, duboko učenje predstavlja autonomni, samo-učeći sustav koji primjenjuje postojeće podatke kako bi našlo uzorke ponašanja i upotrebljava te podatke kako bi predvidjelo nove podatke. Algoritmi dubokog učenja ovo ostvaruju pomoću raznih slojeva umjetnih neuronskih mreža kojima se nastoji oponašati neuroni u mozgu. Na taj način se omogućuje algoritmu da provode razne cikluse kako bi suzio uzorke ponašanja i poboljšao sam sebe svakim ciklusom. U današnjici, primjena dubokog učenja poprima sve veći značaj te primjena sve više raste. U području računalne vizije i raspoznavanju govora ostvaruju čak bolje rezultate od ljudi, a sve veći značaj ima u primjeni autonomnih vozila, koji dubokim učenjem prate okolinu prilikom kretanja te donose odluke tijekom vožnje. [24]

4. UMJETNE NEURONSKE MREŽE

U današnje doba u cijelom svijetu se svakodnevno generira preko 2,5 kvintilijuna bajtova podataka [25] koji se svakodnevno obrađuju digitalnim računalima poprilično velikim brzinama, međutim brzina obrada tih podataka neusporedivo je mala u odnosu na brzinu obrade podataka mozga živih bića. Razlog tome je razlika u građi mozga i računala, gdje mozak ima oko 10^{11} procesora, dok računalo ima 10 ili manje te mozak ima oko 10^3-10^4 veza između procesora, a računalo 10 ili manje [27]. Međutim, iako postoji razlika u građi između mozga i računala, činjenica da ljudska inteligencija predstavlja maksimum inteligencije koliko je danas poznato, želja za ostvarivanjem umjetne inteligencije nalik na ljudsku uopće ne čudi.

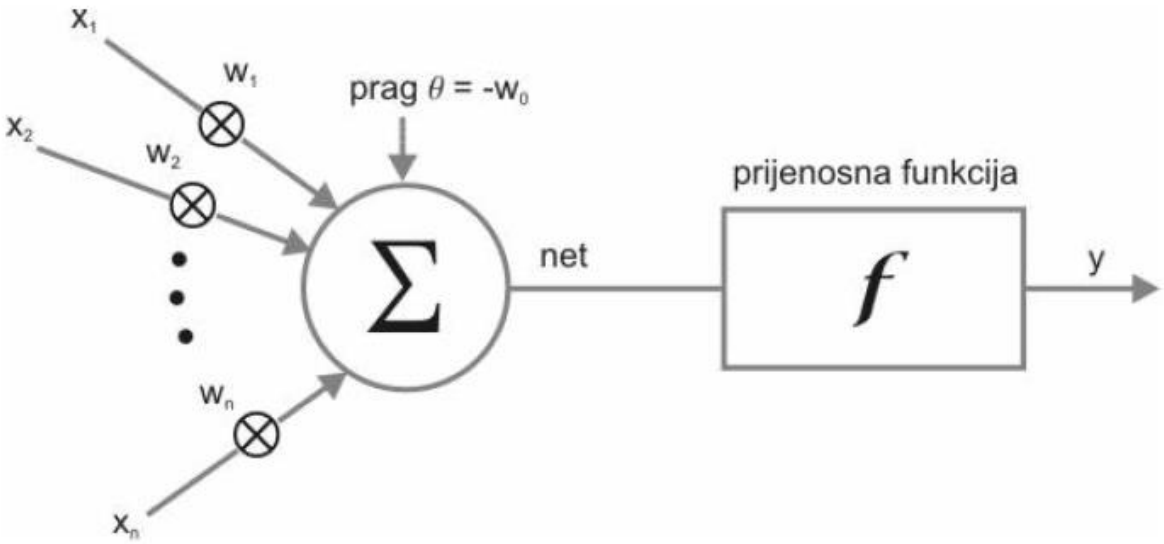
Neuronska mreža predstavlja biološku mrežu građenu od neurona koji su povezani u periferni ili središnji živčani sustav. Neuron predstavlja osnovnu jedinicu živčanog sustava, a ujedno predstavlja i najsloženiju jedinicu ljudskog organizma. Živčani sustav sastoji se od približno 10^{11} međusobno povezanih neurona. Neuron je građen od tijela stanice te mnoštva dendrita i aksona (Slika 15). Dendriti su kraći produžeci koji prenose živčane impulse, dok su aksoni takne cjevčice čiji jedan kraj je povezan s tijelom neurona, dok se drugi kraj grana na niz grana. Svaki neuron je u prosjeku povezan sa 10^4 drugih neurona. Razmak između završetka aksona prethodnog neuorna i dendrita sljedećeg neurona naziva se sinapsa. [26]



Slika 15. Neuron [27]

Neuron signale prima preko dendrita, koji se provode kroz tijelo neurona i predaje ih dalje preko aksona. Zahvaljujući paralelnom radu neurona, mozak postiže visoku brzinu i sposobnost izvršavanja više raznovrsnih zadataka istovremeno.

Ideju o umjetnim neuronskim mrežama, odnosno pokušaju imitacije biološke strukture ljudskog mozga prvi su predstavili 1940-ih godina McCulloch i Pitts (Massachusetts Institute of Technology), istraživajući neurofizičke karakteristike živih bića te objavljivanjem matematičkog modela imena *Threshold Logic Unit (TLU).* Taj model koristi slijedeću analogiju: signali su opisani numeričkim iznosom i na ulazu u neuron množe se težinskim faktorom koja opisuje jakost sinapse; signali pomnoženi težinskim faktorima zatim se sumiraju analogno sumiranju potencijala u tijelu stanice; ako je dobiveni iznos iznad definirana praga, neuron daje izlazni signal [27]. Međutim, mogućnosti tadašnjih računala nisu bila dorasla ovakvom zadatku. Razvojem računalne tehnologije, 1950-ih godina pojavila su se prva praktična ostvarenja, no ubrzo nakon toga, umjetne neuronske mreže pale su u zaborav, sve do 1990. godine te praktički postaju nezaobilazan koncept pri razvoju inteligentnih sustava.



Slika 16. Jednostavni umjetni neuron - perceptron [27]

Na slici 16. vidljiv je najjednostavniji oblik umjetnog neurona – perceptron kojeg je razvio Frank Rosenblatt 1950-ih godina inspiriran radom McCullocha i Pittsa. U strukturi neurona na ulazu su izlazi iz prethodnog neurona, kao i kod bioloških neurona gdje aksoni, odnosno izlazi neurona, povezani na dendrite, odnosno ulaze neurona. Ulazi u neuron označeni sa x_1, x_2,...,x_n, koji se množe sa pripadajućim težinama w_1, w_2,...,w_n te se zbrajaju u tijelu neurona i tako komuniciraju sa neuronom. Zamisao neuronske mreže je da se težine mogu adaptirati i da se mogu naučiti te na taj način mogu kontrolirati utjecaj jednog neurona na drugi. Težinama se zapravo definira važnost značajki. Neuron izračunava prosječni zbroj ulaznih signala i uspoređuje ih sa vrijednošću praga ili bias-a (engl. *thershold*) θ. Pragom se pokušava reflektirati koliko je vjerojatno da je neki uzorak ponašanja prisutan. Ukoliko je ulaz manji od praga,

izlazni neuron poprima vrijednost 0. Ukoliko je ulaz veći ili jednak pragu, neuron se aktivira te izlaz poprima vrijednost 1. Ovo se može prikazati sljedećim formulama (2), (3):

$$net = \sum_{i=1}^{n} W_i \cdot X_i \tag{2}$$

$$f = \begin{cases} 1, & ako\ net\ \geq\ \theta \\ 0, & ako\ net < \theta \end{cases} \tag{3}$$

gdje *net* predstavlja prosječni ulaz u neuron, X_i vrijednost unosa, W_i težinu unosa, *Y* izlaz iz neurona, a θ vrijednost praga. U praksi, ovakav model umjetnog neurona pokazao se nepouzdanim i neučinkovitim zbog otežanog učenja mreže. Problem je uglavnom nastajao zbog diskontinuiteta aktivacijske funkcije čime su nastale prevelike osjetljivosti na male promjene u ulaznim neuronima. Kako bi mreža uspješno učila, iterativnim postupkom potrebno je mijenjati težine, a da bi se to ostvarilo uvedena je kontinuirana aktivacijska funkcija koja se izvodi na zbroju umnožaka ulaza *X* i težina *W*. Aktivacijske funkcije detaljnije će se objasniti u poglavlju 4.2.

4.1. Model umjetne neuronske mreže

Umjetne neuronske mreže nastaju povezivanjem više umjetnih neurona nalik na one spominjane u uvodnom poglavlju umjetnih neuronskih mreža, a način povezivanja neurona određuje arhitekturu mreže. Svaka neuronska mreža sastoji se od 3 sloja: jednog ulaznog sloja i jednog izlaznog sloja te skrivenog sloja. Broj skrivenih slojeva može biti veći od jedan. Ulazni sloj sastoji se od brojeva $x_1,x_2,...,x_D$ koji opisuju neki primjer te se množe sa pripadajućim težinama i pribraja im se prag (bias) u obliku (4):

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} \cdot x_i + w_{j0}^{(1)} \tag{4}$$

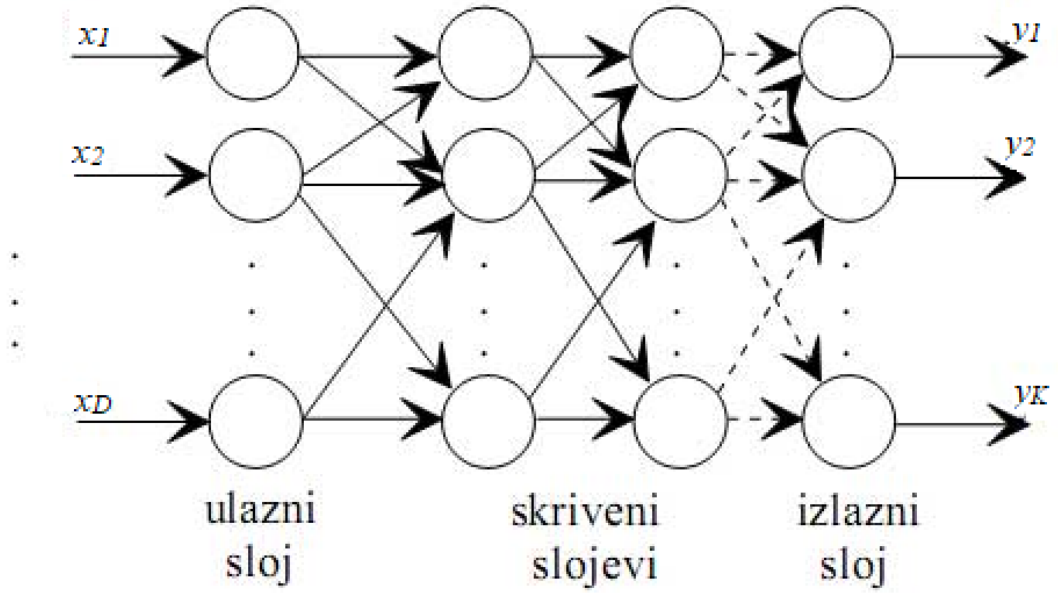
gdje je: $j = 1,...,n$, a *(1)* ukazuje da se radi o prvom sloju neuronske mreže. Veličina a_j naziva se aktivacija, w_{ji} predstavlja težine, a w_{j0} predstavlja prag (bias). Aktivacije a_j se transformiraju nelinearnim aktivacijskim funkcijama $f(\cdot)$ kako bi dale (5):

$$z_j = f(a_j) \tag{5}$$

Izlaskom iz prvog sloja, izlazne vrijednosti se ponovno linearno kombiniraju kako bi dale aktivacije izlaznih jedinica (6).

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} \cdot z_j + w_{k0}^{(2)} \tag{6}$$

gdje je: $k=1,...,K$, a K je ukupni broj izlaza. Oznaka „*(2)*“ ukazuje da se radi o drugom sloju neuronske mreže, a vrijednosti $w_{k0}^{(2)}$ su parametri praga (bias-a). Na kraju, aktivacije izlaznih jedinica se transformiraju primjenjujući prikladnu aktivacijsku funkciju kako bi dali izlaznu vrijednost mreže y_k. [28]



Slika 17. Struktura neuronske mreže [29]

Kombinacijom kroz razne slojeve dobiva se sveukupna funkcija mreže koja se može zapisati kao (7):

$$y_k(x,w) = f\left(\sum_{j=1}^{M} w_{kj}^{(2)} \cdot f\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \qquad (7)$$

Jednadžba (7) predstavlja jednostavni model neuronske mreže koja je zapravo nelinearna funkcija od ulaznih vektora x_i prema izlaznim vektorima y_i, upravljanja zajedničkim vektorom težina i pragova w. Opisana mreža predstavlja unaprijedne propagacije signala, gdje se ulazni signal propagira samo u jednom smjeru kao što je to prikazano na slici 17., a naziv mreže za ovakve slučajeve je unaprijedna ili aciklička neuronska mreža (engl. *Feed-foward Neural Network*). Ovakva mreža naziva se i višeslojni perceptron s obzirom da je nalik na više međusobno povezanih perceptrona pokazanih u uvodnom djelu poglavlja 4. Razlika se javlja u aktivacijskoj funkciji, gdje jednostavni perceptron koristi aktivacijsku funkciju skoka, dok ova neuronska mreža primjenjuje nelinearne aktivacijske funkcije. [28]

Postoje i povratne neuronske mreže (engl. *Reccurent Neural Networks*) koje, za razliku od unaprijednih neuronskih mreža, ne uzimaju kao ulazne informacije samo one koje dolaze iz

prethodnih neurona, već i one podatke koje su obradili u prethodnom koraku. Kombinacijom novih podataka i onih koje su već obradili, mreže donose odluku na temelju više prethodno viđenih podataka. Odluku koju je mreža donijela u trenutku *t-1* utječe na odluku koju mreža donosi trenutak *t* kasnije. To znači da mreža ima dva izvora ulaznih informacija, sadašnju i prošlu informaciju, koje se kombiniraju da bi utvrdila kako reagirati na nove podatke, kao što to rade i ljudski mozak. Upravo je to i glavna razlika u odnosu na unaprijedne mreže kao na slici 18., gdje neuron ponovno uzima svoju izlaznu informaciju. Može se smatrati kako povratna neuronska mreža ima memoriju čime se olakšava obrada sekvencijalnih podataka. Obrađena informacija čuva se u skrivenim jedinicama gdje utječe na obradu novih ulaznih informacija. Tu pronalazi korelacije između ulaza razdvojenih više trenutaka koje se nazivaju „dugoročne korelacije“. [30]

Izlazni podatak neurona u povratnoj neuronskoj mreži može se izraziti sljedećom formulom (8):

$$h_t = f(Wx_t + Uh_{t-1}) \tag{8}$$

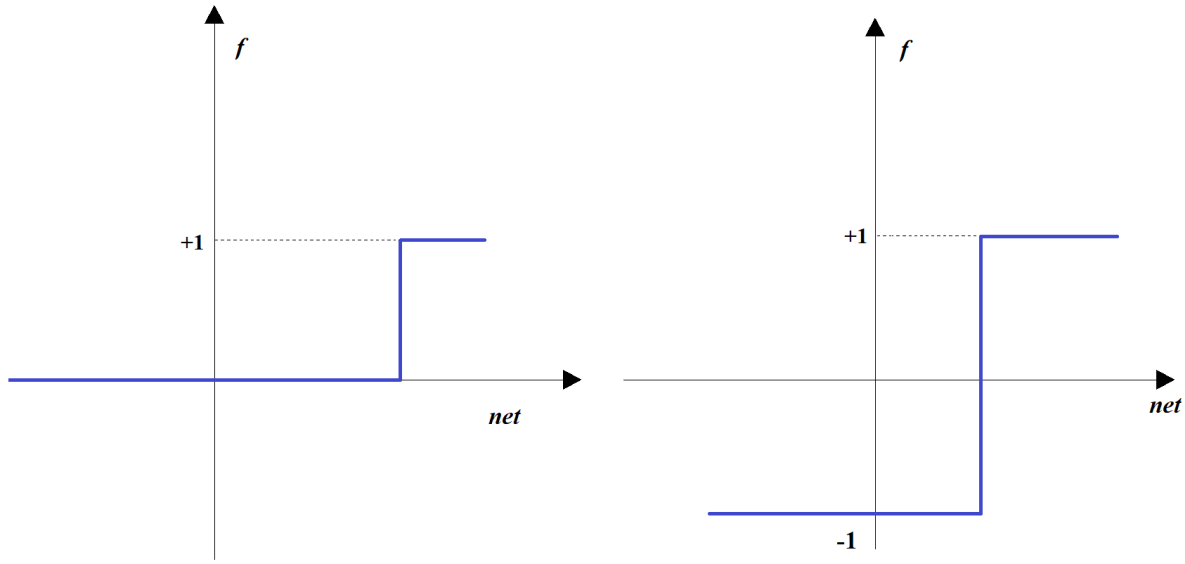
Gdje je h_t skriveno stanje u vremenskom trenutku *t* koje ovisi o ulaznoj informaciji u vremenu *t*, pripadajućoj težini *W* te o skrivenom stanju u vremenskom trenutku *t-1* i pripadajućoj matrici tranzicije *U*. Matrice težina služe kao parametri koji određuju koliku važnost treba pridodati sadašnjem ili prošlom stanju. Greška koja nastaje vraća se algoritmom unazadne propagacije (poglavlje 4.3.3) te se koristi kako bi se prilagodile težine sve dok greška ne dođe na najmanju moguću vrijednost. [30]

4.2. **Aktivacijske funkcije**

Aktivacijska funkcija koristi se kako bi izračunala izlaz iz neurona. Primjenjuju se linearne i nelinearne aktivacijske funkcije. U uvodnom dijelu poglavlja 4 korištena je aktivacijska funkcija imena funkcija skoka ili praga (engl. *Step function*), točnije funkcija binarnog praga koja može imati vrijednosti samo 0 i 1, a ostvaruje se formulom (3). Postoji i bipolarna funkcija praga, koja se razlikuje od funkcije binarnog praga po tome što može primati vrijednosti -1 i 1. Formula je slična kao i formula funkcije binarnog praga, a glasi (9):

$$f = \begin{cases} 1, & ako\ net \geq \theta \\ -1, & ako\ net < \theta \end{cases} \tag{9}$$

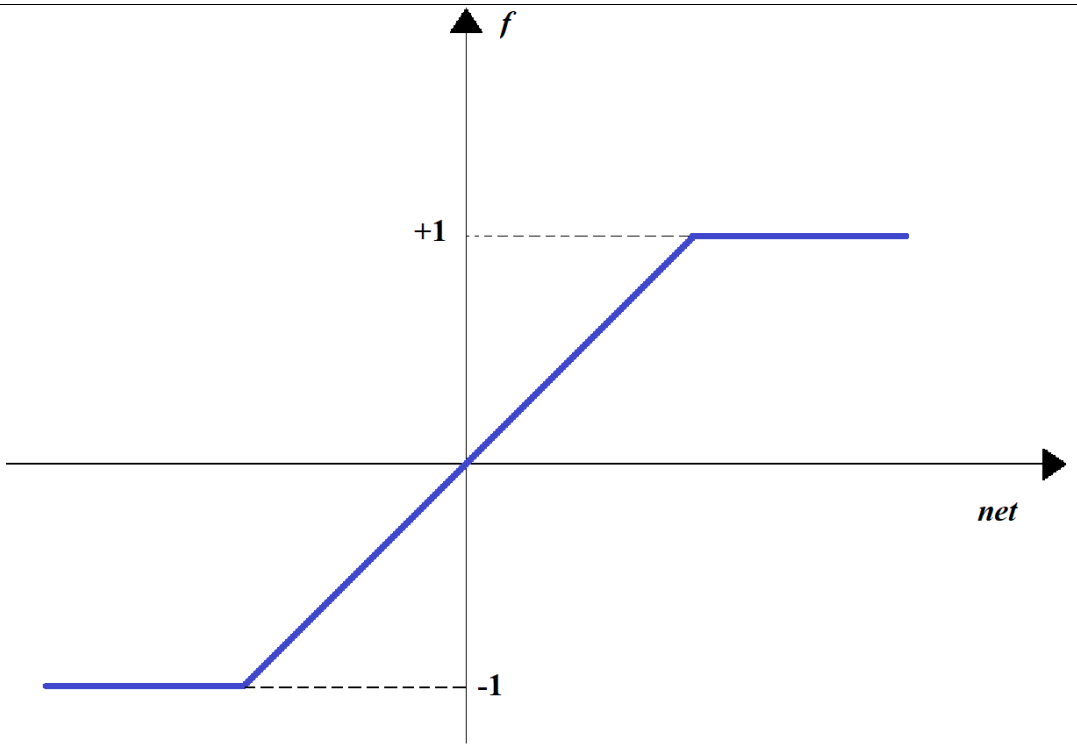
Binarna i bipolarna funkcija praga pokazane su na slici 18.



Slika 18. Binarna (lijevo) i Bipolarna (desno) funkcija praga

Na slici 19. prikazana je djelomična linearna aktivacijska funkcija koja se još naziva i funkcija zasićenja. Radi se o nagnutoj funkciji koja po dijelovima može biti linearna, a opisuje se sljedećim formulama:

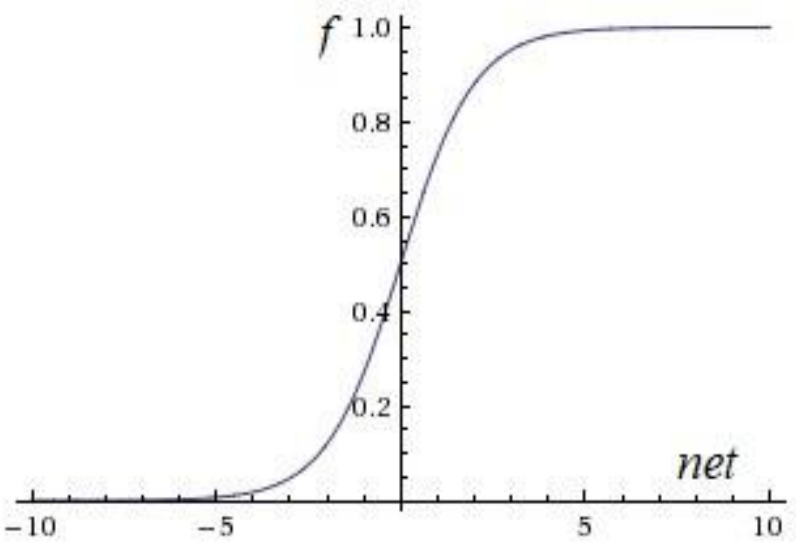
$$f = \begin{cases} -1, & net < a \\ 0, & a \leq net \leq b \\ 1, & net > b \end{cases} \tag{10}$$



Slika 19. Djelomično linearna aktivacijska funkcija

U kategoriju najčešće korištenih aktivacijskih funkcija spada logička funkcija, odnosno sigmoidalna funkcija (slika 20.). Predstavlja nelinearnu aktivacijsku funkciju čija glavna prednost jest što je derivabilna, čime se ostvaruje prednost u postupcima učenja neuronskih mreža pošto se temelji na gradijentnom spustu [27]. Opisuje se formulom (11):

$$f = \frac{1}{1 + e^{-a \cdot net}} \tag{11}$$



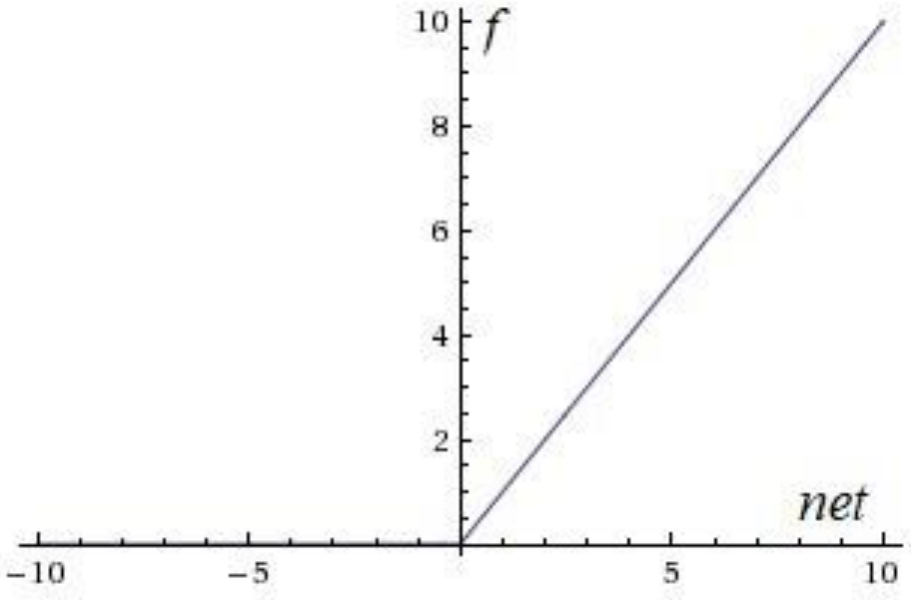
Slika 20. Sigmoidalna funkcija [31]

Funkcija sažima realne brojeve u raspon vrijednosti između 0 i 1. Veliki negativni brojevi poprimaju vrijednost 0, a veliki pozitivni brojevi poprimaju vrijednost 1. U praksi, sigmoidna funkcija nije pokazala zadovoljavajuće rezultate. Korištenjem ove funkcije može se javiti problem nestajućeg gradijenta (o gradijentu je više napisano u poglavlju 4.3.1). To se događa kada neuroni nekog sloja imaju gradijente blizu nuli jer su izlazi tog sloja blizu asimptota. Mreže tog tipa često zapinju u lokalnim minimumima jer mala promjena težina, zbog maloga iznosa gradijenta, uzrokuje sporu konvergenciju [32]. Drugi problem je što su logičke funkcije često su definirane izrazima koje sadrže potenciranje prirodnog logaritma e^x, što je računalno poprilično zahtjevno [33].

Rectified linear unit (*ReLU*) je nedavno postala najkorištenija aktivacijska funkcija, opisana sljedećim izrazom (12):

$$f(net) = \max(0, net) = \begin{cases} 0, ako\ je\ net < 0 \\ net, ako\ je\ net \geq 0 \end{cases} \quad (12)$$

ReLU funkcijom rješavaju se navedeni problemi koji se javljaju kod sigmoidnih funkcija. Međutim, i ova funkcija ima svoje mane. Ukoliko neki neuron nije aktivan, gradijent je 0, što znači da se težine tog neurona više ne mogu pomaknuti gradijentnom optimizacijom. Ovo se može izbjeći na način da umjesto da funkcija poprima vrijednost 0 ako je *net<0,* ona poprima vrijednost malu vrijednost (npr. 0,01), čime bi svi neuroni bili aktivni. ReLU funkcija prikazana je na slici 21.



Slika 21. ReLU aktivacijska funkcija [31]

Softmax funkcija izračunava raspodjelu vjerojatnosti događaja preko *n* različitih događaja. Drugim riječima, ovom funkcijom izračunava se vjerojatnost ciljne klase nad svim ostalim mogućim ciljnim klasama. Glavna značajka ove funkcije je raspon vjerojatnosti izlaza, sve vrijednosti izlaza bit će između 0 i 1 a njihov ukupan zbroj treba biti 1. Ukoliko se funkcija koristi na više klasifikacijskom modelu, funkcija vraća vjerojatnost svake klase, a ciljna klasa bi trebala imati najveću vjerojatnost. [34]

Softmax funkcija se računa sljedećom formulom (13):

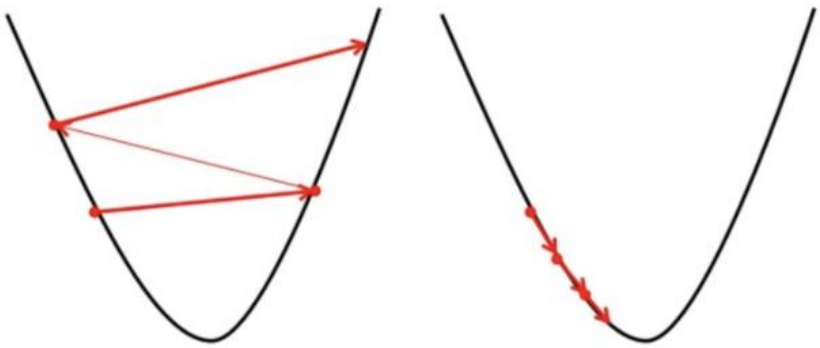
$$f = \frac{e^z}{\sum e^z} \tag{13}$$

4.3. Učenje neuronskih mreža

Kao što je u uvodnom poglavlju 4. rečeno, strojevi svoju inteligenciju stječu kroz učenje, a ovdje će se objasniti kako se uz pomoć neuronskih mreža postiže učenje. Kako bi neuronska mreža mogla rješavati stvarne probleme potrebno je podešavanje njezinih parametara. Za podešavanje parametara primjenjuju se dvije tehnike: gradijentni spust (engl. *Gradient Descent*) i algoritam propagacije unatrag (engl. *Backpropagation*).

4.3.1. Gradijentni spust

Gradijentni spust predstavlja optimizacijski algoritam čiji je cilj pronaći minimum derivabilne konveksne funkcije. Lokalni minimum funkcije traži se na način da se trenutnoj točki nekakve konveksne funkcije traži smjer gradijenta funkcije te se u sljedećem koraku kreće u smjeru maksimuma negativnog gradijenta. Međutim, bitno je da korak koji se napravi bude dovoljno malen kako bi nova vrijednost točke bila manja od prethodne. Ukoliko se napravi preveliki korak postoji mogućnost da će se točka pomaknuti na suprotnu padinu funkcije i zapravo povećati vrijednost funkcije u toj novoj točki. Ovo se može vidjeti na slici 22.



Slika 22. Veliki korak (lijevo), mali korak (desno) [35]

Kod učenja neuronskih mreža funkcija koja se optimira je funkcija gubitaka *E*, a kako bi se algoritam gradijentnog spusta mogao koristiti za podešavanje težina neuronske mreže, potrebno je da ta funkcija gubitaka bude derivabilna. [33]

Iz navedenoga se može zaključiti kako je potrebno pronaći vektor težina *w* koji će minimizirati funkciju cilja, odnosno funkciju gubitaka *E(w)*. Prvo, napravi se mali korak u promjeni iznosa težine *w* na sljedeći način (14):

$$w = w + \delta w \tag{14}$$

Promjenom iznosa težina mijenja se i funkcija gubitaka *E* (15)*:*

$$\delta E \cong \delta w^T \nabla E(w) \tag{15}$$

Gdje je *∇E(w)* gradijentni vektor funkcije gubitaka te predstavlja smjer u kojem se ostvaruje najveći pad funkcije gubitaka. Najmanja vrijednost funkcije gubitaka ostvarit će se na mjestu gdje (16):

$$\nabla E(w) = 0 \tag{16}$$

Prethodnu formulu (16) gotovo je nemoguće ostvariti analitičkim jednadžbama jer funkcija gubitaka ovisi o mnogo parametara (težina i pragova), stoga se primjenjuju iterativni postupci. Uglavnom, iterativne tehnike provode se na način da se izabire nasumična vrijednost $w^{(0)}$ za vektor težina te se pomiče kroz prostor na sljedeći način:

$$w^{(\tau+1)} = w^{(\tau)} + \Delta w^{(\tau)} \tag{17}$$

, gdje τ označuje iteracijski korak.

Najjednostavniji pristup optimiziranju parametara je pomicanje u malim koracima i to u smjeru suprotnom od smjera vektora gradijenta funkcije gubitaka. Ovo se može zapisati u obliku formule (18), što predstavlja formulu (17) zapisanu na drukčiji način:

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E\left(w^{(\tau)}\right) \tag{18}$$

, gdje η predstavlja stopu učenja. Nakon svakog ažuriranja, gradijent se ponovno vrednuje za novi vektor težine te se postupak ponavlja. Ovaj postupak se isto ovako provodi i za pragove *b*. [28]

4.3.2. Stohastički gradijentni spust

U slučajevima kada se upravlja većim količinama podatka, algoritam gradijentnog sputa pokazao se neučinkovitim. Gradijentnim spustom, algoritam prolazi kroz skup podataka za treniranje te se provodi podešavanje za svaki primjer koristeći jednadžbu (18). Potrebno je nekoliko prolaza kroz skup podataka za treniranje dok se algoritam ne konvergira, a učestala je

primjena miješanja skupa podataka za učenje prije svakog prolaza što dodatno usporava postupak učenja.

Poduzorkovanjem skupa za treniranje sa nasumično odabranim podacima zvanim *mini-batch* može se aproksimirati ∇E te se time ubrzava proces treniranja neuronske mreže. Upravo to se naziva stohastički gradijentni spust. Daljnji se proces treniranja mreže provodi sa *mini-batch* podacima kao ulaznim vrijednostima te se proces ponavlja sve dok se ne iscrpe podaci na kojima se izvršava treniranje, što se naziva epoha (engl. *epoch*). [36]

4.3.3. Algoritam unazadne propagacije

Algoritam unazadne propagacije prvi put je predstavljen 1970-ih godina, no popularnost je stekao tek 1986. godine kada su Rumelhart, Hinton i Williams objavili rad imena "Learning Representations by Back-Propagating Errors" kojim je opisano nekoliko tada poznatih neuronskih mreža i učenje na njima propagacijom unatrag. Učenje propagacijom unatrag bilo je brže od do tada poznatih metoda učenja. Danas ova metoda predstavlja najkorišteniju metodu učenja. [37]

Srž algoritma propagacije unatrag predstavlja parcijalna derivacija funkcije gubitaka E u odnosu na težine w ili pragove b unutar mreže kojom se pokazuje koliko brzo se funkcija gubitaka mijenja u odnosu na težine (ili pragove).

$$\frac{\partial E}{\partial w} \;\; ili \;\; \frac{\partial E}{\partial b} \tag{19}$$

Kako bi se bolje razumio algoritam propagacije unatrag moraju se definirati određeni pojmovi vezani uz njega. w^l_{jk} označava težinu za vezu k-tog neurona u $(l\text{-}1)$ sloju prema j-tom neuronu u sloju l. Slično označavanje primjenjuje se i kod pragova i aktivacija. Tako prag b^l_j pripada j-tom neuronu u sloju l, a kod aktivacija a^l_j pripada j-tom neuronu u l-tom sloju. Primjenjujući ove oznake, aktivacija a^l_j j-tog neurona u l-tom sloju povezanu sa aktivacijama u $(l\text{-}1)$-tom sloju se formulom može zapisati kao (20):

$$a^l_j = h\left(\sum_k w^l_{jk} a^{l-1}_k + b^l_j\right) \tag{20}$$

Prije pojašnjavanja algoritma unazadne propagacije također treba napomenuti dvije pretpostavke koje idu uz algoritam. Prva pretpostavka je da funkcija gubitaka E može biti zapisana kao prosjek funkcija gubitaka E_x za individualni slučaj, x. To se može zapisati na sljedeći način (21):

$$E = \frac{1}{n}\sum_x E_x \tag{21}$$

Na ovakav način, kvadratna funkcija troškova može se zapisati na način (22):

$$E_x = \frac{1}{2}\|x - a^L\|^2 \tag{22}$$

Razlog zbog kojeg nam je ova pretpostavka bitna je što taj algoritam propagacije unatrag omogućuje da izračunamo parcijalne derivacije $\partial E_x/\partial w$ *i* $\partial E_x/\partial b$ za jedinstvene trening slučajeve. Drugom pretpostavkom traži da se funkcija gubitaka može zapisati kao funkcija izlaza iz neuronske mreže. Kao primjer se može uzeti kvadratna funkcija gubitaka na način (23):

$$E = \frac{1}{2}\|x - a^L\|^2 = \frac{1}{2}\sum_j \left(y_j - a_j^L\right)^2 \tag{23}$$

, gdje *y* predstavlja željenu izlaznu vrijednost, a vrijednost *L* označava broj slojeva unutar mreže.

Kako bi se mogle računati parcijalne derivacije navedene formulom (19), mora se uvesti vrijednost greške δ. Algoritam propagacije unatrag bazira se na 4 osnovne jednadžbe. Te jednadžbe zajedno daju mogućnost računanja greške i gradijenta funkcije gubitaka. Prva jednadžba je jednadžba za računanje greške u izlaznom sloju, δ^L (24):

$$\delta_j^L = \frac{\partial E}{\partial a_j^L} \cdot h'\left(z_j^L\right) \tag{24}$$

, gdje $\partial E/\partial a^L_j$ mjeri kako se brzo mijenja funkcija *j*-te izlazne aktivacije, $h'(z^L_j)$ mjeri koliko se brzo aktivacijska funkcija mijenja u z^L_j, a *z* predstavlja malu promjenu težine neurona. Jednadžba (24) može se i zapisati u obliku pogodnijem za matrice (25)

$$\delta^L = \nabla_a E \odot h'(z^L) \tag{25}$$

, gdje se $\nabla_a E$ se definira kao vektor čije komponente su parcijalne derivacije $\partial E/\partial a^L_j$, a „⊙“ označava elementarni produkt dva vektora. Druga jednadžba na kojoj je baziran algoritam propagacije unatrag je jednadžba za grešku δ^l u pogledu na idući sloj, δ^{l+1} (26)

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot h'(z^l) \tag{26}$$

, gdje $(w^{l+1})^T$ predstavlja transponiranu matricu težina w^{l+1} za sloj *(l+1)*. Ako se pretpostavi da se zna greška δ^{l+1} za sloj *(l+1)* te se primjeni transponirana matrica težina $(w^{l+1})^T$, onda se intuitivno može misliti kako se pogreška kreće unatrag kroz mrežu, dajući mjeru pogreške na izlaznom *l* sloju. Sa produktom $\odot h'(z^l)$, greška se pomiče unatrag preko aktivacijske funkcije u sloju *l*, dajući pogrešku δ^l. Jednadžbama (24) ((25) za matrice) i (26) može se izračunati greška za svaki δ^l sloj u mreži.

Treća jednadžba na kojoj je baziran algoritam propagacije unatrag je jednadžba za stopu promjene funkcije gubitaka s obzirom na bilo kakav prag u mreži (27):

$$\frac{\partial E}{\partial b_j^l} = \delta_j^l \tag{27}$$

Ova jednadžba (27) pokazuje da je stopa promjene $\partial E/\partial b^l_j$ jednaka grešci δ^l_j.

Posljednja jednadžba na kojoj je baziran algoritam propagacije unatrag je jednadžba stope promjene funkcije gubitaka s obzirom na bilo koju težinu u mreži (28):

$$\frac{\partial E}{\partial w} = a_k^{l-1}\delta_j^l \tag{28}$$

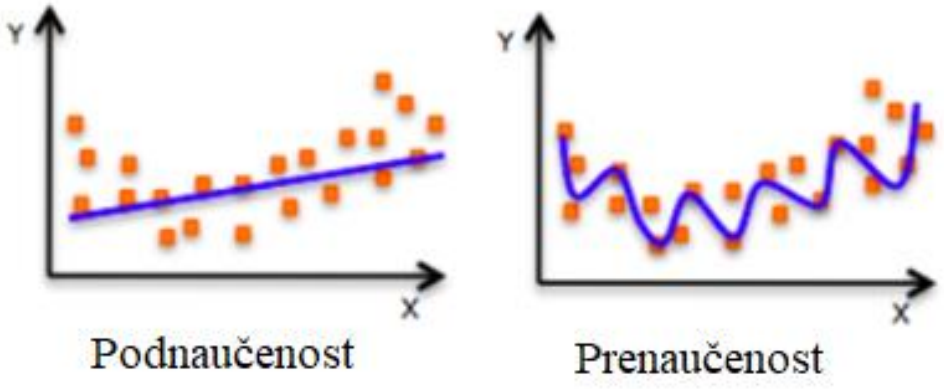
Kao što je već rečeno, algoritmom propagacije unatrag računa se gradijent funkcije gubitaka za svaki set trening podataka. Algoritam se provodi kroz sljedeće korake:

1. Ulazni podaci *x*: Postavlja se odgovarajuća aktivacijska funkcija a^1 za ulazi sloj.
2. Provodi se algoritam propagacije unaprijed: za svaki $l=2,3,...,L$ računa se $z^l = w^l a^{l-1} + b^l$ i $a^l = h(z^l)$.
3. Izlazna greška δ^L: Računa se vektor $\delta^L = \nabla_a E \odot h'(z^L)$.
4. Propagacija greške unatrag: Za svaki $l=L-1, L-2,...,2$ računa se $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot h'(z^l)$.
5. Izlaz: Gradijent funkcije gubitaka dobiva se: $\frac{\partial E}{\partial w^l_{jk}} = a_k^{l-1}\delta_j^l$ i $\frac{\partial E}{\partial b_j^l} = \delta_j^l$.

[38]

4.4. Regularizacija neuronskih mreža

Sposobnost neuronskih mreža da točno predviđa izlazne vrijednosti podataka sa kojima se još nije susrela predstavlja njihov glavni cilj. Sposobnost algoritma u obavljanju zadataka na do tada neviđenim ulazima (podacima) naziva se *generalizacija*. Kao što je u poglavlju 4.3 rečeno, neuronske mreže uče smanjujući grešku, tj. funkciju gubitaka. Jedan od problema koji može nastati tijekom učenja neuronskih mreža je prenaučenost (engl. *overfitting*). U slučaju prenapučenosti greška koja nastaje učenjem neuronske mreže na trening setu (setu podatak na kojem mreža uči) podataka je mala, međutim kada se unesu novi, nepoznati podaci, greška znatno poraste. To znači da se javio problem u kojem je mreža memorirala trening podatke, ali nije sposobna rješavati zadatke na novom skupu podataka. Također, može doći i do problema podnaučenosti (engl. *underfitting*), gdje model niti ne može naučiti model na trening setu podataka, samim time niti točno rješavati nove zadatke. Na slici 23. prikazano je učenje modela uz prisutnost prethodno navedenih problema.



Slika 23. Utjecaj podnaučenosti i prenaučenosti na sposobnost generalizacije [39]

Sa slike 23. odmah je vidljivo kakav problem se javlja kod podnaučenosti, dok se kod prenaučenosti na prvi pogled može se smatrati kako se jako dobro naučio model, odnosno predobro jer u podacima na kojima model uči često su prisutni razni šumovi koje je, u ovom slučaju, model naučio kao ulazne podatke. Upravo to predstavlja jedan od razloga zbog kojeg model daje lošija predviđanja na novim ulaznim podacima. Jedan od načina rješavanja ovih problema je jednostavno povećanje trening seta podataka ili smanjenje veličine mreže. Međutim, veće mreže su potencijalno i moćnije nego manje, stoga to nije povoljna opcija koju će se olako odabrati.

Ovi problemi zato se mogu riješiti metodom regularizacije algoritma. Regularizacija se najčešće primjenjuje na funkciju gubitaka tako što se uvodi kažnjavanje visokih vrijednosti težina (engl. *weight decay*). [39]

$$E = E_0 + \frac{\lambda}{2n}\sum_{w} w^2 \tag{29}$$

Na početnu, neregulariziranu funkciju gubitaka E_0 dodaje se suma kvadrata težina te se skalira faktorom $\lambda/2n$, gdje λ predstavlja parametar regularizacije, a kao n se najčešće uzima veličina trening seta podataka. [38]

4.5. Konvolucijske neuronske mreže

U posljednjih nekoliko godina, umjetna inteligencija sve više i više uspijeva nadmašiti ljude u različitim izazovima iz različitih domena. Iako se može smatrati da umjetna inteligencija ima prednost nad ljudima u raznim područjima kao što je računanje, u posljednje vrijeme ostvaruje prednost i u područjima koje ljudi izvršavaju bez previše truda, dok je za umjetnu inteligenciju to kompleksan problem. Primjer tome je prepoznavanje lica na slikama, operacija koju čovjek izvršava u djeliću sekunde, a za računalnu viziju to predstavlja kompleksan zadatak. Do nedavno se umjetna inteligencija nije mogla ni uspoređivati sa ljudima u takvom zadatku, no razvojem neuronskih mreža, točnije konvolucijskih neuronskih mreža, ona uspijeva čak i nadmašiti ljude. 1998. godine Le Cun, Bengio, Bottou i Haffner [40] objavljuju rad o konvolucijskoj neuronskoj mreži u kojoj opisuju arhitekturu prve konvolucijske neuronske mreže LeNet-5 koja je mogla klasificirati ručno pisane znamenke. 2012. godine Alex Krizhevsky [41] objavljuje dublju i proširenu verziju LeNet mreže te je s tom mrežom je pobijedio na ImageNet natjecanju.

Kao što je navedeno u poglavlju 4.1, neuronske mreže se mogu dijeliti u slojeve (unutarnji, skriveni i vanjski) te se u svakom od njih nalazi više neurona. Unutar istog sloja, svaki od tih neurona prepoznaje slične uzorke. Ukoliko ima više skrivenih slojeva, neuroni u prvom skrivenom sloju (skriveni sloj bliži ulaznom sloju) prepoznaju značajke niže razine kao što su linije i rubovi, a prolazeći sve dublje kroz slojeve, prepoznaju se značajke sve viših razina kao što su primjerice oči, uši, usta, u mrežama čiji je zadatak prepoznavanje lica. Treniranjem mreže podešavaju se težine i pragovi koji predstavljaju parametre mreže, a pojam hiperparametar predstavlja atribute kojima se određuju postavke treniranja te se oni moraju ručno podesiti.

Kada se razvija sustav dubokog strojnog učenja, preporuča se da sustav uči i pokreće modele na „sirovim“ podacima, bez izvlačenja značajki ručno [42]. Razlog tome je da bi mreža mogla trenirati na „sirovim“ podacima i kako bi sama mogla naučiti izvlačiti značajke. Međutim, kada bi se uzela slika koja se sastoji od velike količine piksela te je svaki od tih piksela je označen raznim vrijednostima koje prezentiraju boju, prezentacija te slike u ulazni sloj neuronske mreže postaje kompleksna. Današnji standard slike predstavlja slika sa 1920x1080 piksela što znači da bi ulazni sloj neuronske mreže trebao imati preko 2 milijuna neurona. Kako se neuronskom mrežom podešavaju parametri (težine), ova operacija bi bila poprilično vremenski zahtjevna. Kako bi se riješio takav problem, razvijene se konvolucijske neuronske mreže. One su specifično primijenjene za rad sa slikama. Iz tog razloga su konvolucijske neuronske mreže organizirane u 3 dimenzije: visina, širina i dubina. Dubina predstavlja dimenziju koja razlikuje

različite vrijednosti boje. Uz to, konvolucijske neuronske mreže dodaju novi skriveni slojevi: konvolucijski sloj i sloj sažimanja. Kao aktivacijska funkcija u konvolucijskim neuronskim mrežama primjenjuje se ReLU, a vrsta mreže koja se primjenjuje u konvolucijskim neuronskim mrežama je unaprijedna propagacijska mreža (engl. *Feed-foward Network*) sa povratnim prostiranjem pogreške. [42]

Kao što je prethodno navedeno konvolucijske mreže primjenjuju se pretežito za slike. Jednostavno rečeno, svaka slika predstavlja raspored piksela raspoređenih u određenom redoslijedu. Ako se promijeni red ili broj piksela, promijenit će se i slika. Računalo upravo tu sliku pretvara u matricu piksela, odnosno zapisuje ih kao dvodimenzionalnu matricu brojeva. Konvolucijskim neuronskim mrežama zadatak je zapravo shvatiti raspored i uzorak tih brojeva. Znajući ovo, može se zaključiti da za primjenu konvolucijskih neuronskih mreža nije nužno da je ulazni podatak slika. Signal dobiven vibracijama također predstavlja matricu brojeva te će se upravo ovim radom konvolucijska neuronska mreža primijeniti na vibracijske signale ležajeva.

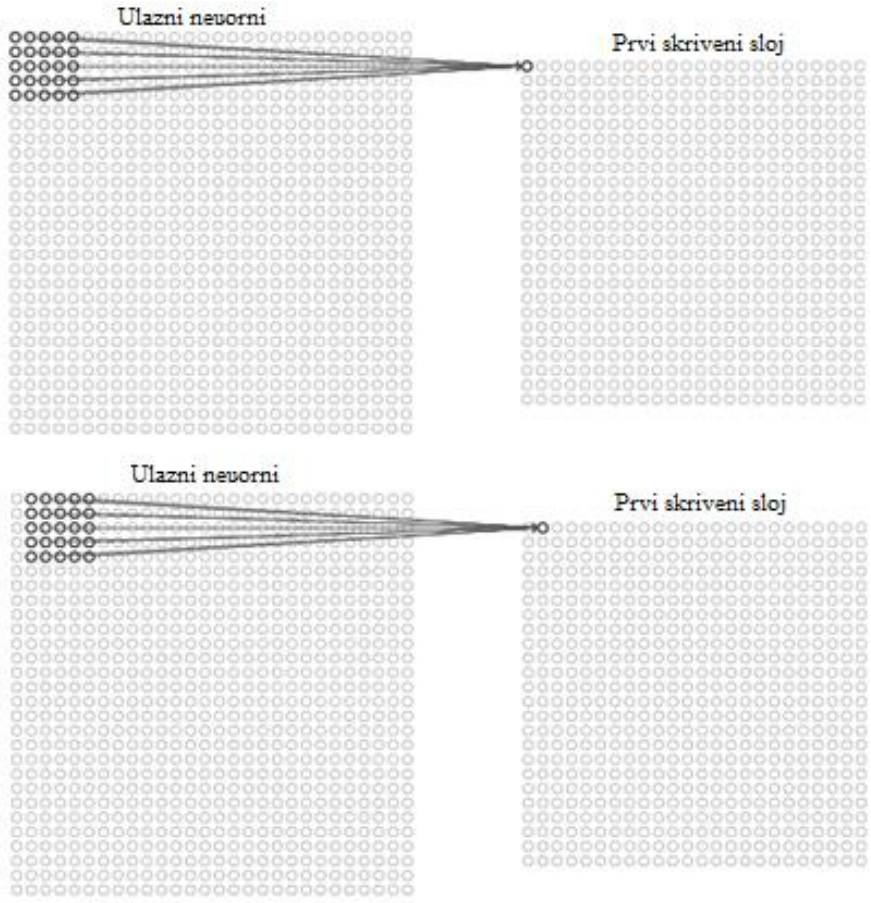
Konvolucijski sloj

Kako bi se nosili sa problemima sa kojima se suočavaju neuronske mreže sa potpuno međusobno povezanim slojevima, primjenjuju se konvolucijski slojevi. Ono što razlikuje konvolucijske mreže od običnih neuronskih mreža jest to da se svi neuroni ulaznog sloja ne povezuju sa svim neuronima prvog skrivenog sloja, već se manja grupa neurona iz ulaznog sloja povezuje s neuronom izlaznog sloja kao što se to može vidjeti na slici 24. Pojam konvolucije zapravo se odnosi na matematičku kombinaciju dviju funkcija kako bi se proizvela treća funkcija, odnosno time se spajaju dva seta informacija.

Da bi se započeo postupak konvolucije, uzima se polje kvadratnog oblika sa širinom i duljinom kao dimenzijama. Te dimenzije predstavljaju hiperparametar zvan *receptivno polje*, a te dimenzije moraju biti jednake dimenziji filtera. Što se tiče dimenzije dubine, to nije potrebno definirati, s obzirom da se konvolucije same po sebi provode kroz sve dubine. Razlog zbog kojeg se dubina kao dimenzija definira u ulaznim podacima je taj da bi se definirale različite boje koje je obično potrebno kombinirati kako bi se izvukle korisne informacije. [42]

Konvolucijska operacija provodi se na način da se matrica receptivnog polja množi sa matricom filtera. Filter (engl. *kernel*) predstavlja set naučenih parametara (težina i pragova) koji se dobiju u svakom koraku pomicanja receptivnog polja. Na slici 24. prikazan je primjer ulaznog sloja koji predstavlja sliku dimenzija 28x28 piksela, što znači da ulazni sloj se sastoji od 784 neurona, a dimenzije receptivnog polja su postavljene na dimenzije 5x5. Tih 25 neurona određenih receptivnim poljem iz ulaznog sloja množi se s matricom filtera te se rezultat pohranjuje u

obliku jednog neurona u idući sloj baš kao na slici 24. Ovaj postupak se dalje ponavlja kroz cijeli ulazni sloj kao što se vidi na slici 24. Korak kojim se kreće po ulaznom sloju predstavlja hiperparametar, tj. vrijednost koju ručno određujemo i postavljamo. U primjeru sa slike 24., vrijednost koraka kretanja iznosi jedan korak.

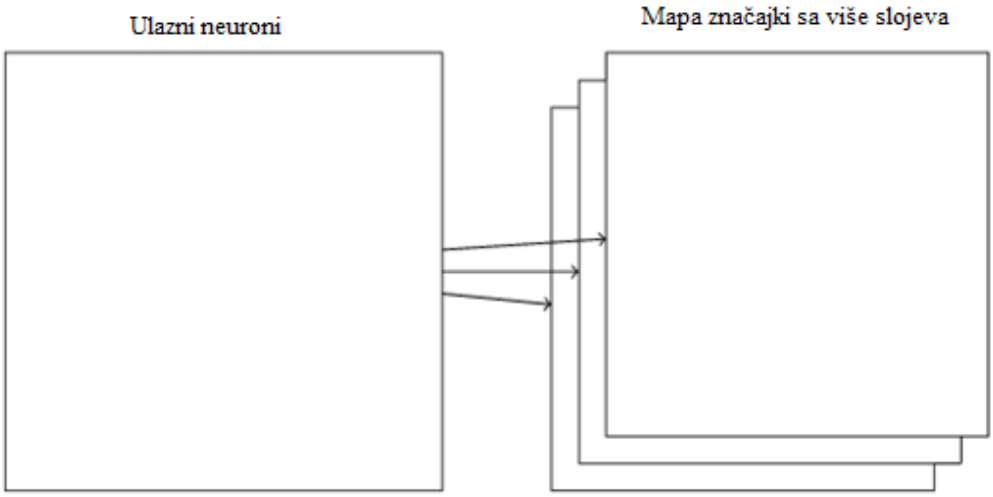


Slika 24. Konvolucijska operacija sa receptivnim poljem 5x5 [38]

Nakon učenja mreže, vrijednosti težina i pragova koji povezuju polja ulaznih i skrivenih neurona imat će iste vrijednosti. Ovo predstavlja karakteristiku konvolucijskih neuronskih mreža zvanu *dijeljenje parametara* (engl. *weight sharing*). 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. [13]

To omogućuje da konvolucijske neuronske mreže uče razne značajke, od najjednostavnijih pa do kompleksnijih. Iz tog razloga se skriveni sloj može se nazvati mapom značajki te nije ograničen samo na jednu značajku. Unutar jednog skrivenog sloja, moguće je naučiti više mapi značajki, od čega svaka mapa ima zaseban naučeni set parametara (slika 25.). Ovime se ostvaruje trodimenzionalnost mreže što predstavlja jednu od najbitnijih razlika između konvolucijskih neuronskih mreža i običnih neuronskih mreža. Ulazne datoteke mogu biti

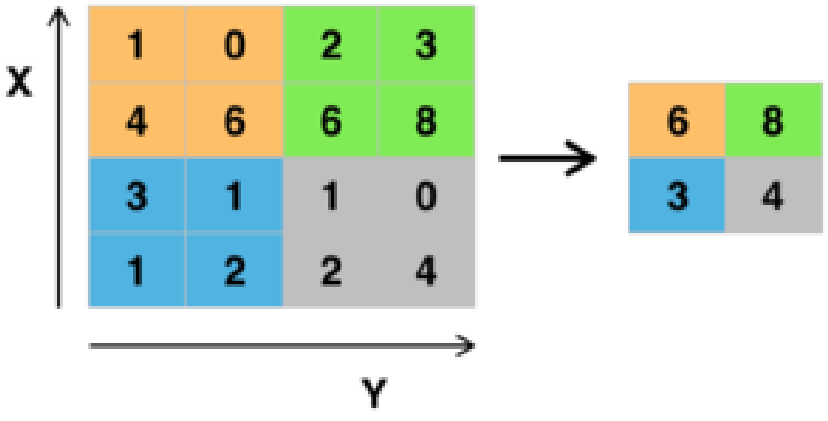
sačinjene od više slojeva, a isto tako i skriveni slojevi mogu imati više mapi značajki kao na slici 25., čime se omogućuje učenje raznih značajki. [38]



Slika 25. Skriveni sloj sa 3 mapi značajki [38]

Sloj sažimanja

Sloj sažimanja predstavlja treći sloj u konvolucijskim mrežama, nakon ulaznog i konvolucijskog, čiji je cilj smanjivanje složenosti konvolucijskih neuronskih mreža i sprječavanje prenaučenosti mreže. Slično konvolucijskom sloju, neuroni u sloju sažimanja su povezani u pravokutnik sa dimenzijama širine i duljine. Slojem sažimanja se skupine susjednih piksela zamjenjuju prosječnom vrijednosti neurona skupine, ako se radi o prosječnom sažimanju (engl. *average pooling*), ili maksimalnom vrijednosti, ako se radi o maksimalnom sažimanju (engl. *max pooling*). Primjer sažimanja dan je na slici 26.



Slika 26. Postupak sažimanja (maksimalno sažimanje) [43]

Misao koja stoji iza sažimanja je da u konvolucijskoj neuronskoj mreži nije toliko bitna lokacija značajke, već samo njena relativna pozicija u odnosu na druge značajke. Sloj sažimanja imat će jednaku dubinu kao i konvolucijski sloj koji sažima.

Batch normalizacija

Kada se trenira, neuronska mreža se želi normalizirati. Normalizacijom se nastoji transformirati podatke kako bi se svi podaci mogli postaviti na istu skalu. Tipičan postupak normalizacije skalira numeričke vrijednosti da budu u vrijednostima između 0 i 1. Jedan od razloga za uvođenje normalizacije je mogućnost postojanja situacije u kojoj bi jedna težina bila mnogo veća od ostalih težina, samim time i izlaz iz neurona povezanog sa tom vezom bi bio velik te bi se stvorila neravnoteža. Stoga se uvodi sloj *batch normalizacije*. U prvom koraku, podaci koji ulaze u sloj normalizacije se normaliziraju preko formule (30):

$$z = \frac{x - m}{s} \tag{30}$$

Gdje je x vrijednost podatka, m prosječna vrijednost podatka te s standardna devijacija tog skupa podataka. Dalje, ova vrijednost z se množi novom proizvoljno odabranom vrijednosti g.

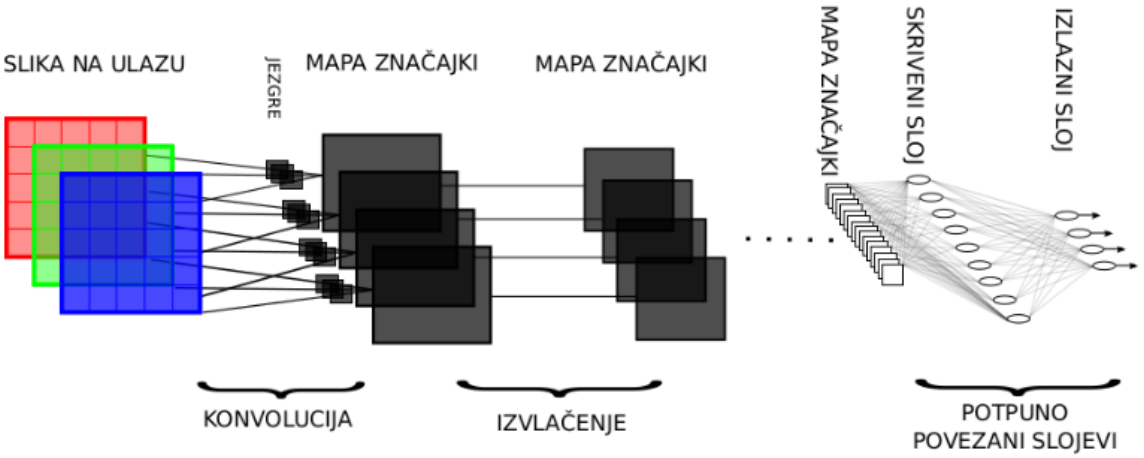
$$z \cdot g \tag{31}$$

Na kraju se formula (31) zbraja sa proizvoljnom vrijednosti l.

$$(z \cdot g) + l \tag{32}$$

Vrijednosti m, s, g, l su vrijednosti koje se mogu trenirati, što znači da se optimiziraju učenjem mreže. Ovime se ostvaruje da težine unutar mreže ne postanu nebalansirane.

Primjer konvolucijske neuronske mreže



Slika 27. Primjer strukture konvolucijske neuronske mreže [44]

Iz ulaznog sloja provodi se operacija konvolucije kojom se dobiva prvi skriveni sloj, kao što je opisano u prethodnom dijelu. U konvolucijskim slojevima izabiru se hiperparametri, kao što su veličina i broj filtera čime se određuje mapa značajki. Primjenom lokalnih receptorskih polja, postiže se smanjenje broja parametara koje mreža mora naučiti te se smanjuje broj računskih operacija koje mreža mora naučiti.

Tablica 1. Parametri konvolucijskog sloja

Konvolucijski sloj	
Ulazne dimenzije	$W_1 \times H_1 \times D_1$
Izlazne dimenzije	$W_2 \times H_2 \times D_2$
	$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
	$H_2 = W_2$ za slučaj 2D ulaza, $H_2 = 1$ za slučaj 1D ulaza
	$D_2 = K$
Hiperparametri	Broj filtera, K
	Veličina filtera, F
	Korak filtera, S
	Nadopuna, P

Nakon što su dobivene mape značajki, potrebno je, kao i običnim neuronskim mrežama, provesti nelinearne aktivacijske funkcije. Najčešće primjenjivana aktivacijska funkcija je ReLU funkcija objašnjena u poglavlju 4.2. Nakon aktivacijske funkcije slijedi sloj sažimanja kojim se nastoji smanjiti dimenzionalnost prethodnog sloja kao što je objašnjeno u prethodnom dijelu.

Tablica 2. Parametri sloja sažimanja

Sloj sažimanja	
Ulazne dimenzije	$W_2 \times H_2 \times D_2$
Izlazne dimenzije	$W_3 \times H_3 \times D_3$
	$W_3 = \frac{(W_2 - F)}{S} + 1$
	$H_3 = W_3$ za slučaj 2D ulaza, $H_3 = 1$ za slučaj 1D ulaza
	$D_3 = D_2$
Hiperparametri	Veličina filtera, F
	Korak filtera, S

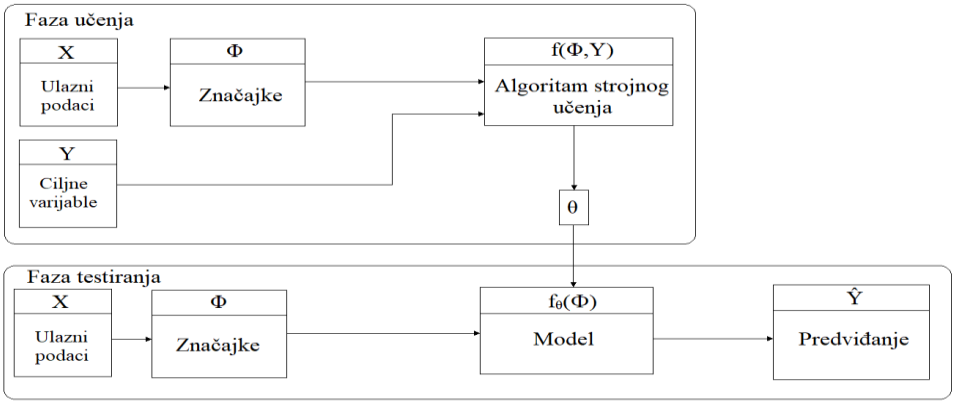
Na kraju konvolucijske neuronske mreže nalazi se jedan ili više povezanih slojeva koji zapravo predstavljaju slojeve iz obične neuronske mreže.

5. PROCJENA UZROKA I INTENZITETA OŠTEĆENJA LEŽAJEVA

Izbjegavanje zastoja u radu strojeva, smanjenje troškova generiranih neadekvatnim održavanjem, produljenje životnog vijeka opreme i povećanje sigurnosti njihovog rada spadaju među ključne aktivnosti održavanja. Ovim radom će se primijeniti strategija održavanja po stanju primjenom algoritama dubokog učenja.

Strojno učenje izvlačenjem značajki

Održavanjem po stanju može se smatrati donošenje odluka o potrebnim zahvatima održavanja, potpomognuto računalnom procjenom stanja opreme. Računalna procjena stanja opreme može se ostvariti algoritmima strojnog učenja koji imaju mogućnost prepoznavanja određenih stanja opreme. Kako bi algoritam strojnog učenja mogao prepoznavati razna moguća stanja potrebno je njegovo učenje preko značajki. Prvo, uzima se set ulaznih podataka mjerenja sa označenim ciljnim varijablama odnosno oznakama različitih stanja koje ti ulazni podaci predstavljaju. Zatim, algoritam uči, korištenjem izvučenih značajki iz prethodno definiranog seta podataka kako bi naučio različita stanja tijekom mjerenja. Ovaj proces vidljiv je na slici 28., gdje se u fazi učenja ulazni podaci X skupljaju zajedno sa odgovarajućom ciljanom varijablom Y. Iz podataka izvlače se značajke Φ koje se dodjeljuju algoritmu učenja zajedno sa ciljnim varijablama Y. Algoritam strojnog učenja preko izvučenih značajki Φ i ciljnih varijabli Y uči model θ kojim može razlikovat razna stanja. Kada se algoritam dovoljno dobro nauči, u fazi testiranja uzimaju se novi ulazni podaci kojima algoritam još nije imao pristupa te se iz njih izvlače značajke i unose se u model θ koji preko znanja koja je naučio u fazi učenja ima mogućnost predviđanja stanja stroja ili opreme $\hat{Y}$ novih podataka. Za izvlačenje značajki iz mjerenja potrebni su stručnjaci za praćenje stanja opreme koji će određivati koje značajke se izvlače.

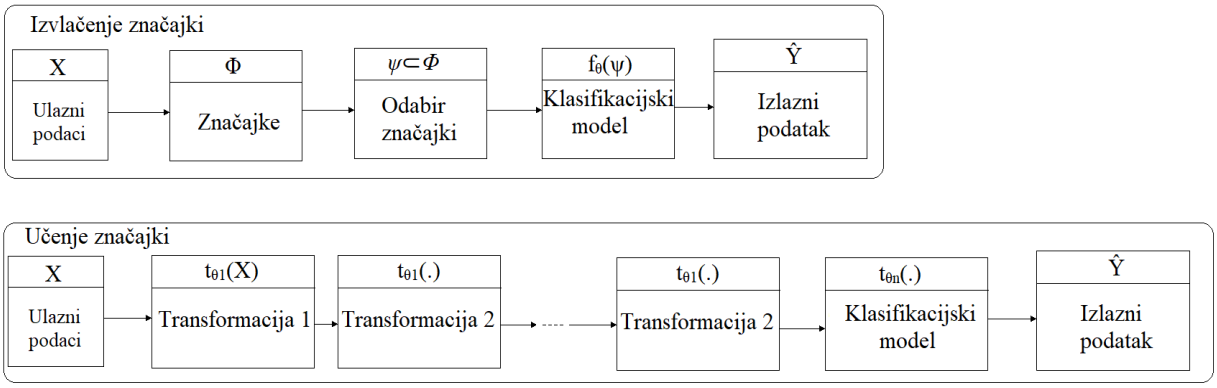


Slika 28. Postupak određivanja stanja opreme izvlačenjem podataka i algoritmom strojnog učenja [45]

Međutim, postoji više razloga da prethodno naveden sustav izvlačenja značajki neće davati optimalne rezultate: (1) izvlačenje značajki ovisi o stručnjaku i njegovom znanju o strojarstvu ili statistici koji definira koje se značajke izvlače, (2) mogućnost da potrebno znanje za stvaranje značajki još nije dostupno, (3) efikasni su i točni uglavnom ukoliko je dostupni matematički opis modela. Također, sustav praćenja stanja može biti specificiran za klasifikaciju određenog seta stanja, pa samim time pojavom novih stanja sustav neće davati adekvatnu klasifikaciju.

Strojno učenje učenjem značajki (duboko učenje)

Kako bi se riješili prethodno navedeni problmeni, primjenjuje se učenje značajki. Za razliku od izvlačenja značajki gdje stručnjak određuje značajke koje se izvlače, učenjem značajki primjenjuje algoritam dubokog učenja sam uči i generira značajke iz neobrađenih podataka. Na slici 29. dan je shematski prikaz učenja značajki i usporedba sa shemom izvlačenja značajki.



Slika 29. Razlika između izvlačenja značajki i učenja značajki [45]

Učenjem značajki ne izvlače se značajke, već se primjenjuje transformaciju $t_{\theta 1}(.)$ na neobrađene podatke, gdje se θ_1 se sastoji od parametara koji se mogu naučiti tijekom transformacije. Transformacijom se dobiva novi prikaz ulaznih podataka, koji su onda prikladniji za generiranje značajki te se transformacija može ponavljati sve dok sustav optimalno ne nauči značajke za klasifikacijski zadatak. Među najpoznatije metode učenja značajki spadaju algoritmi dubokog učenja. [45]

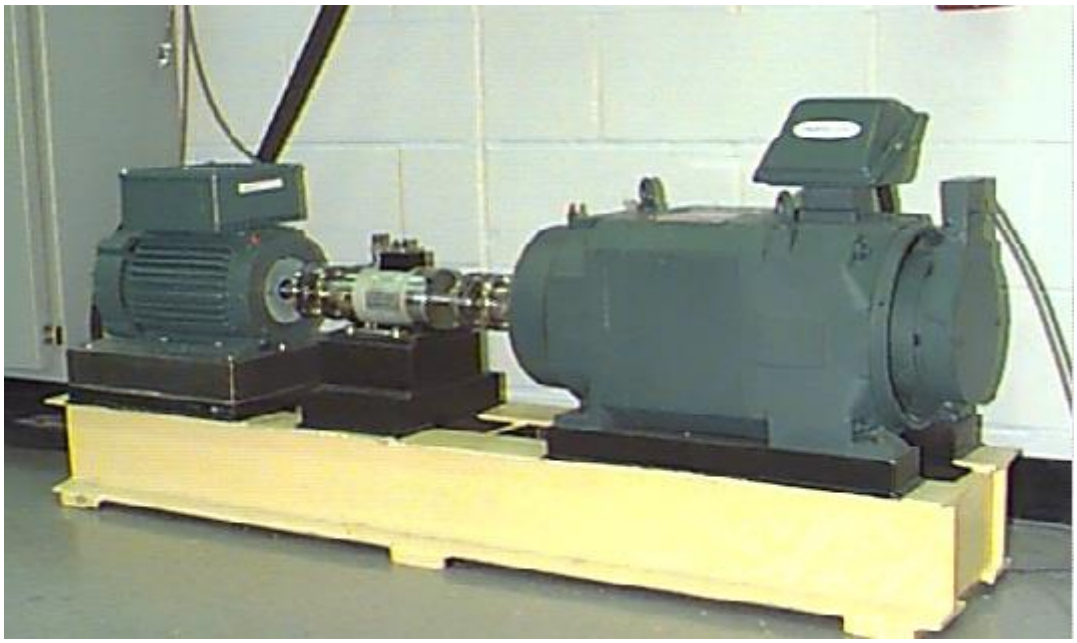
Algoritmima dubokog učenja omogućava se modeliranje višedimenzionalnih reprezentacija podataka te klasificiranje ili predviđanje stanja stroja ili opreme. Zasniva se na činjenici da prolaskom ulaznih podataka kroz višestruke slojeve mreže, model samostalno obrađuje podatke te generira korisne značajke. Značajke se generiraju polazeći od najjednostavnijih pa svakim idućim slojem do sve kompleksnije reprezentacije ulaznog signala. [46]

Uz prethodno navedene prednosti u korištenju algoritama dubokog učenja kod praćenja stanja strojeva i opreme, u sklopu diplomskog rada primijenit će se algoritam konvolucijskih

neuronskih mreža opisan u poglavlju 4.5 u svrhu identifikacije vrste kvara na kugličnim ležajevima uz pomoć vibracijskih podataka.

5.1. Ulazni set podataka

Postupak procjene uzroka kvarova konvolucijskim neuronskim mrežama provest će se na podacima kugličnih ležajeva dostupnih na web stranici Sveučilišta Case Western Reserve [47]. Podaci koji će se koristiti za učenje mreže sastoje od podataka dviju pozicija na kojima su postavljani ležajevi sa raznim oštećenjima te u jednom slučaju ležaj bez oštećenja i nad njima su snimane njihove vibracije tijekom rada. Prvi ležaj nalazi se na pogonskoj strani motora (na slici 30. lijevo), a drugi ležaj se nalazi kod ventilatora (na slici 30. desno).



Slika 30. Ispitna stanica [47]

Na slici 30. vidi se ispitna stanica na kojoj je provedeno ispitivanje ležajeva. Stanica se sastoji od motora sa 2 KS (na slici lijevo), pretvarača momenta (sredina), dinamometra (desno) te električnog upravljača. Ležajevi koji su se ispitivali podupirali su vratilo motora. Greške dobivene na ležajevima na kojima je proveden eksperiment dobivene su elektroerozijskom obradom (EDM). Ležajevi koji će se koristiti u ovom radu sastoje se od oštećenja na kotrljajućim elementima, na unutarnjem prstenu te na vanjskom prstenu i to sa različitim veličinama promjera oštećenja: 0,007inch (≈0,18mm), 0,014inch (≈0,36mm), 0,021inch (≈0,53mm). Uz ove oštećene ležajeve uzet će se i vibracijski podaci neoštećenog ležaja. Specifikacije ležajeva koji će se koristiti u radu su sljedeće:

Tablica 3. Specifikacije ležaja na pogonskoj strani motora

Pozicija	Pogonska strana motora
Naziv	6205-2RS JEM
Proizvođač	SKF
Vrsta ležaja	Kuglični ležaj s dubokim utorima
Promjer unutarnjeg prstena	0,9843inch (≈25mm)
Promjer vanjskog prstena:	2,0472inch (≈52mm)
Debljina:	0,5906inch (≈15mm)
Promjer kuglice:	0,3126inch (≈7,94mm)
Promjer između kugličnih elemenata:	1,537inch (≈39,04mm)

Tablica 4. Specifikacije ležaja na poziciji ventilatora

Pozicija	Ventilator
Naziv	6203-2RS JEM
Proizvođač	SKF
Vrsta ležaja	Kuglični ležaj s dubokim utorima
Promjer unutarnjeg prstena	0,6693inch (≈17mm)
Promjer vanjskog prstena:	1,5748inch (≈40mm)
Debljina:	0,4724inch (≈12mm)
Promjer kuglice:	0,2656inch (≈6,75mm
Promjer između kugličnih elemenata:	1,122inch (≈28,5mm)

Podaci koji će se primjenjivati u radu predstavljaju vibracijske signale dobivene akcelerometrima koji su na kućište bili pričvršćivani magnetima i to na poziciji od 90° (6 sati) za ležaj pogonskog motora te na kućištu ventilatora. Signali su prikupljani preko 16-kanalnog akvizicijskog sustava te obrađeni u programskom sučelju MATLAB. Podaci su skupljani u 12000 uzoraka u sekundi, broj okretaja motora iznosio je 1797 okretaja po minuti, a akcelerometri su prikupljali vrijednosti vibracija za 1 os. [47]

Na ovom setu podataka već su provedena neka istraživanja u klasifikaciji oštećenja. Primjerice u [48] također je korištena konvolucijska neuronska mreža u klasifikaciji oštećenja, bez izvlačenja značajki iz signala, gdje je korištena mreža sa 6 konvolucijskih i sažimajućih slojeva te su postignuti rezultati u točnosti od 99,75 ± 0,08%.

5.2. Programsko sučelje

Kao alat kojim će se primijeniti algoritam konvolucijska neuronska mreža u ovom radu je MATLAB te njegov paket „*Deep Learning Tooloox*“.

Program MATLAB služi za rješavanje različitih matematičkih problema te čitav niz izračuna i simulacija vezanih uz obradu signala, upravljanje, regulaciju i identifikaciju sustava. Prva verzija MATLAB-a, jednostavni „matrični laboratorij“ (Matrix Laboratory), napisana je krajem 1970. godine na sveučilištima University of New Mexico i Stanford University s ciljem primjene u matričnoj teoriji, linearnoj algebri i numeričkoj analizi. Korišten je Fortran i dijelovi biblioteka LINPACK i EISPACK. Početkom 80-tih se prelazi na C programski jezik uz dodavanje novih mogućnosti, i to prvenstveno u područjima obradbe signala i automatskog upravljanja. Od 1984. MATLAB je dostupan kao komercijalni proizvod tvrtke MathWorks [49].

Danas svojstva MATLAB-a daleko prelaze originalni “matrični laboratorij”. Radi se o interaktivnom sustavu i programskom jeziku za opća tehničke i znanstvene izračune. Osim osnovnog sustava postoje i brojni programski paketi koji ga proširuju te pokrivaju gotovo sva područja inženjerske djelatnosti: obradu signala i slike, 2D i 3D grafičke prikaze, automatsko upravljanje, identifikaciju sustava, statističke obrade, analizu u vremenskoj i frekvencijskoj domeni, simboličku matematiku i brojne druge. Jedan od važnijih paketa je SIMULINK— vizualni alat koji omogućuje simulaciju kontinuiranih i diskretnih sustava pomoću funkcijskih blok dijagrama te time ne zahtijeva od korisnika detaljno poznavanje sintakse nekog programskog jezika. MATLAB je također zamišljen kao sustav u kojem korisnik na jednostavan način može graditi svoje vlastite alate i biblioteke te modificirati postojeće. [49]

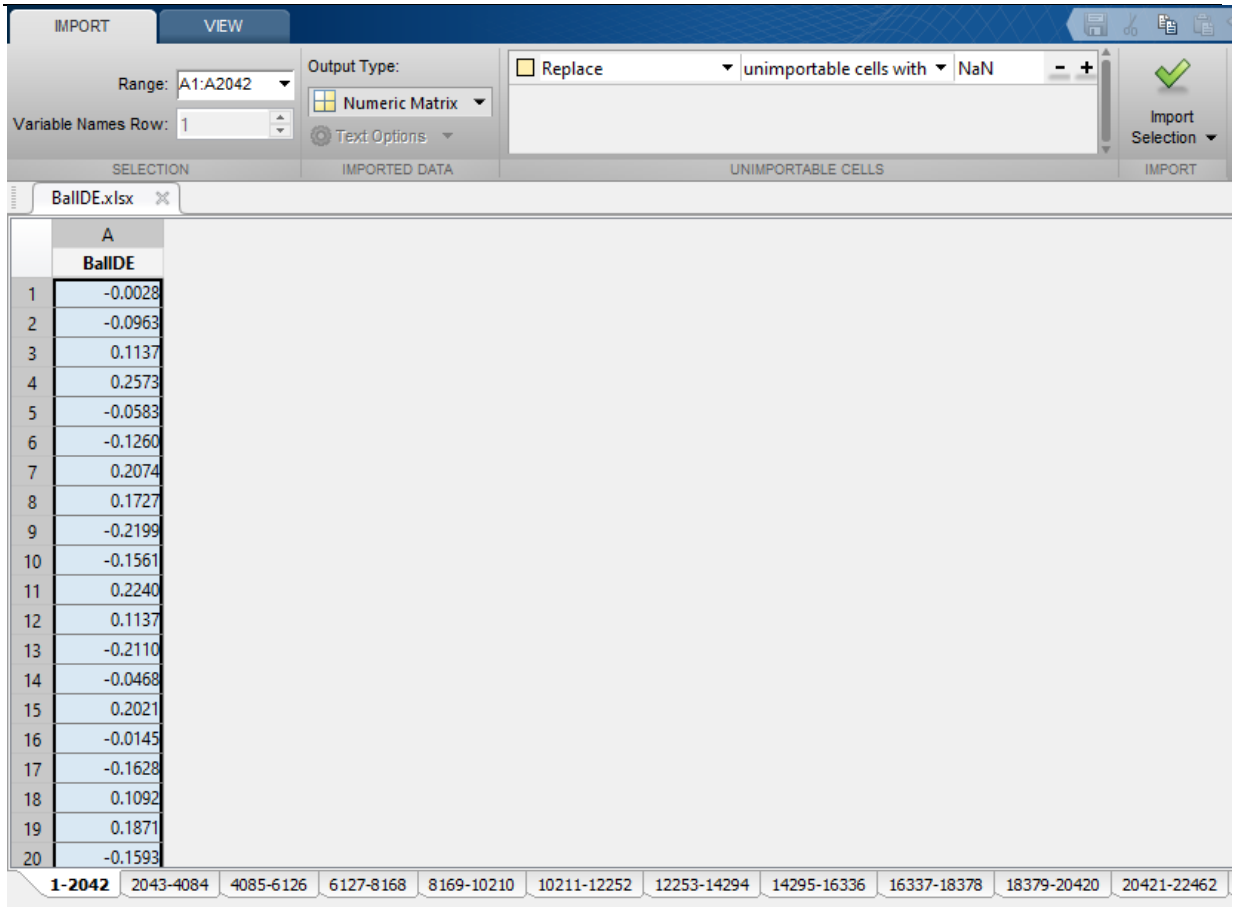
Kao što je prethodno navedeno, u radu će se primjenjivati „*Deep Learning Toolbox“*. Ono pruža okvir za projektiranje i implementaciju dubokih neuronskih mreža s algoritmima, prethodno istreniranim modelima i aplikacijama. Omogućuje korištenje konvolucijskih neuronskih mreža i drugih kratkoročno memorijskih mreža za klasifikaciju i regresiju na slikama, vremenskim serijama i tekstualnim podacima.

5.3. Učitavanje ulaznih podataka

Prvi korak u izradi konvolucijskih neuronskih mreža je učitavanje ulaznih podataka kojima bi mreža mogla pristupiti za trening te za testiranje i validaciju naučene mreže. Podaci koji će se koristiti kao što je već navedeno preuzeti su sa web adrese [47]. Radi se o .mat datotekama za sve vrste kvarova navedenih u poglavlju 5.1, a unutar svake datoteke nalaze podaci za obje vrste ležaja (obje pozicije ležajeva) i vrijednost rotacijske brzine motora koji su također spomenuti u poglavlju 5.1. Prije samog učitavanja podataka, kako bi omogućili rad konvolucijske neuronske mreže, potrebno je pripremiti podatke. Podaci o svakom ležaju imaju oko 120 000 zapisa vibracijskih signala, a kako bi se omogućilo učenje neuronske mreže dijeljenjem na trening i test set podataka, ovih 120 000 zapisa podijelit će se na 60 dijelova što znači da će svaki set podataka sadržavat oko 2000 zapisa vibracijskih signala. Za dijeljenje podataka koristit će se program „Microsoft Excel“ u koji će se za svaki ležaj prebaciti zapisi vibracijskih signala te podijeliti na 60 dijelova i spremiti u obliku .xlsx (*Microsoft Excel Spreadsheets*). Kako bi se učitale vrijednosti vibracijskih signala korištena je MATLAB funkcija Import te su sve vrijednosti unošene u obliku matrica (slika 31.). Kao rezultat učitavanja dobiva se stupac koji sadrži amplitude u vremenskoj domeni (vibracijski signali). Ovaj stupac sprema se u posebnu .mat datoteku.

Cilj ovog rada je procjena uzroka i intenziteta oštećenja ležaja, a s obzirom da postoji veći broj dostupnih mjerenja, predloženi sustav obradit će se na način da se podaci sa više senzora za pojedini tip kvara grupiraju na način da predstavljaju više dimenzija. U ovom slučaju koristit će se vibracijski podaci ležajeva sa različitih pozicija (ležaj na pogonskoj strani motora i ležaj ventilatora) koji imaju jednake kvarove te će se oni grupirati u višedimenzionalnu matricu.

Grupiranjem se dobiva matrica dimenzija 2042x1x2 što znači da ulazni podaci u konvolucijsku neuronsku mrežu sastoje 2 seta podataka od po 2042 vibracijskih podataka u jednom stupcu, odnosno ulazne podatke na kojima će mreža učiti i testirati se predstavlja matrica koja sadrži podatke 2 različita ležaja ali sa istim oštećenjem.



Slika 31. Učitavanje podataka unutar MATLAB-a

5.4. Konvolucijska neuronska mreža i odabir hiperparametara

Nakon što su učitani ulazni podaci na kojima se provodi trening, test i validacija, slijedi izrada i treniranje konvolucijske neuronske mreže. Kao što je već navedeno u uvodnom dijelu poglavlja 5., algoritmima dubokog učenja nije potrebno izvlačiti značajke iz ulaznih podataka stoga nije potrebna nikakva daljnja obrada niti priprema podataka. Ovim radom pokazat će se efikasnost konvolucijskih neuronskih mreža u procjeni kvarova ležajeva, točnije u klasifikaciji kvarova ležajeva navedenih u poglavlju 5.1. Analizirajući podatke vidljivo je da će mreža se sastojat od 10 neurona, s obzirom da se radi sa podacima o greškama ležajeva na tri različite pozicije te su za svaku od tih grešaka dana su tri različita promjera te greške. Podaci ovih grešaka bit će pohranjeni u 10 različitih mapa na koje će se pozivati algoritam tijekom učenja mreže. Ovaj korak naime predstavlja i idući korak u izradi konvolucijskih neuronskih mreža nakon što se učitaju podaci.

Nadalje, nad učitanim mjerenjima izvršava se podjela na trening skup, test skup i validacijski skup. Trening skup predstavlja podatke koje će mreža vidjeti i nad kojima će učiti, validacijski skup služi za nepristranu procjenu modela tijekom učenja mreže, a test skup služi za provjeru

generalizacije neuronske mreže, odnosno to su podaci kojima mreža izvršava procjenu modela a da im tijekom učenja mreža nije imala pristupa.

Nakon što su učitani podaci, definirane njihove mape i podijeljeni na trening, test i validacijski skup slijedi izrada strukture konvolucijske neuronske mreže i postavljanje njihovih hiperparametara. Način provjere uspješnosti rada mreže provodi se jednostavnom metodom pokušaja i pogrešaka.

Struktura koja je odabrana u ovom radu sastoji se od 2 skupa slojeva, gdje svaki skup ima jedan konvolucijski sloj, aktivacijski sloj i sloj sažimanja. U prvom skupu slojeva, konvolucijski sloj sastoji se od 32 filtera kojim se uče karakteristične težine na način spomenut u poglavlju 4.5 i stvara 32 mape značajki. Nakon konvolucijskog sloja postavljen je sloj batch normalizacije. Nakon sloja batch normalizacije slijedi aktivacijski sloj. Kao aktivacijska funkcija u ovom sloju odabrana je ReLU funkcija opisana u poglavlju 4.2. Kako bi se smanjila dimenzionalnost mreže, nakon aktivacijskog sloja postavljen je sloj sažimanja, sa maksimalnim sažimanjem. Prikaz slojeva dan je na slici 32. te u tablici 5. su prikazani definirani parametri.

```
 1   'input'            Image Input             2042x1x2 images with 'zerocenter' normalization
 2   'conv_1'           Convolution             32 56x1 convolutions with stride [1  1]
 3   'batch_1'          Batch Normalization     Batch normalization
 4   'relu_1'           ReLU                    ReLU
 5   'max_pool_1'       Max Pooling             2x1 max pooling with stride [2  2]
 6   'conv_2'           Convolution             32 10x1 convolutions with stride [1  1]
 7   'batch_2'          Batch Normalization     Batch normalization
 8   'relu_2'           ReLU                    ReLU
 9   'max_pool_2'       Max Pooling             2x1 max pooling with stride [2  2]
10   'fc'               Fully Connected         10 fully connected layer
11   'softmax'          Softmax                 softmax
12   'classification'   Classification Output   crossentropyex
```

Slika 32. Slojevi konvolucijske mreže

Tablica 5. Struktura konvolucijske neuronske mreže i njezini parametri

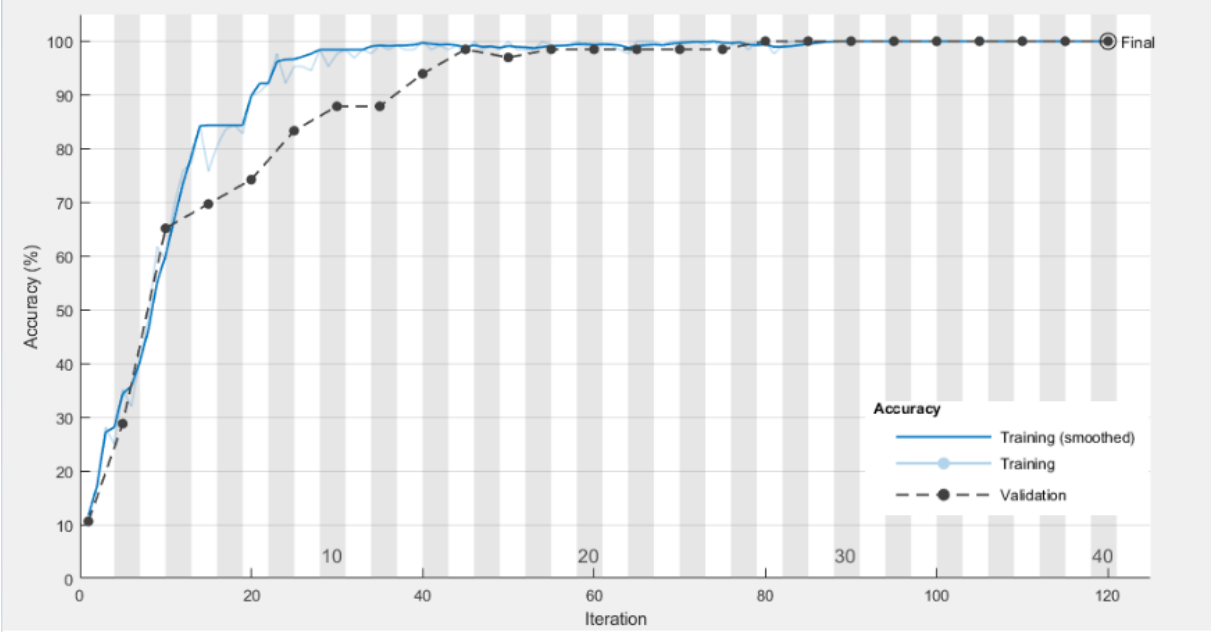
Sloj	Parametri i veličina sloja
Ulazni sloj	Ulazni signal: [2042 x 1 x 2]
Konvolucijski sloj 1	32 filtera veličine [56 x 1 x 2] Veličina sloja: [1987 x 1 x 32]
Batch normalizacija 1	Normalizacija
Aktivacijski sloj 1	ReLU
Sažimajući sloj 1	Maksimalno sažimanje [2 x 1] Veličina sloja [994 x 1 x 32]
Konvolucijski sloj 2	32 filtera veličine: [10 x 1 x 32] Veličina sloja: [963 x 1 x 32]
Batch normalizacija 2	Normalizacija
Aktivacijski sloj 2	ReLU
Sažimajući sloj 2	Maksimalno sažimanje [2 x 1]
Potpuno povezani sloj	Veličina sloja: 10
Izlazni sloj	Klasifikacija

Nakon što se definiraju slojevi i parametri konvolucijske mreže, potrebno je definirati i metodu i parametre učenja te mreže. Kao način učenja mreže odabran je stohastički gradijentni spust. Kao maksimalan broj epoha uzeta je vrijednost 40, a za stopu učenja odabrana je vrijednost 0,0007. Također, kako bi se bolje naučio model mreže odabrano je miješanje podataka prije svake epohe. U tablici 6. pokazani su parametri učenja.

Tablica 6. Parametri učenja konvolucijske neuronske mreže

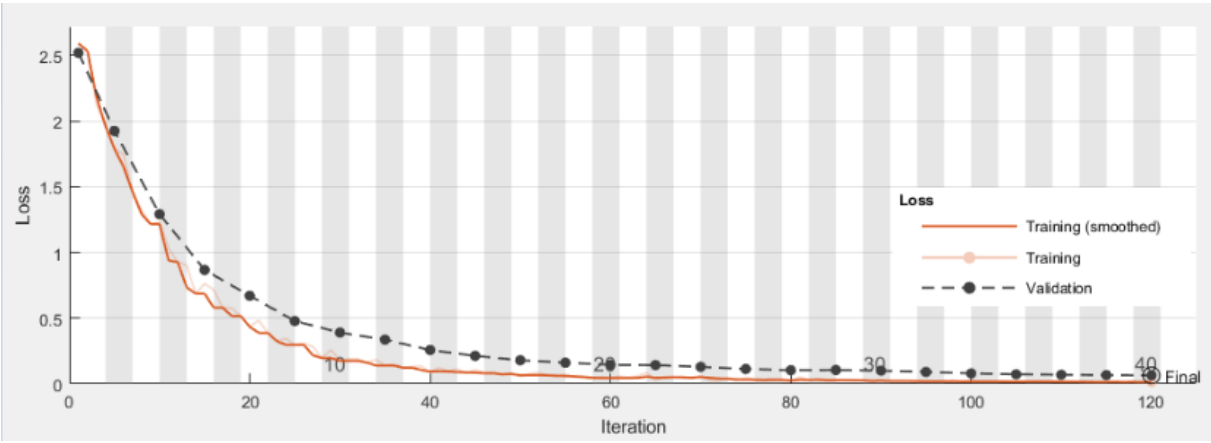
Parametri učenja	
Algoritam učenja	Stohastički gradijentni spust
Stopa učenja	0,0007
Miješanje podataka	Da, svaku epohu
Maksimalan broj epoha	40
Moment	Da

Na slici 33. može se vidjeti grafički prikaz učenja mreže.



Slika 33. Proces učenja konvolucijske neuronske mreže

Na slici 34. prikazana je funkcija gubitaka tijekom učenja.



Slika 34. Funkcija gubitaka tijekom učenja konvolucijske neuronske mreže

5.5. Rezultati konvolucijske neuronske mreže

Tijekom učenja mreže, provest će se 3 slučaja kojima će se trenirati mreža kako bi se vidjelo koja opcija daje najbolje rezultate. U prvom slučaju koristit će se 75% podataka kao trening test, 15% podataka kao test set i 10% podatka kao validacijski set podataka. U drugom slučaju 60% podatka činit će trening set, 30% test set podatka i 10% kao validacijski set. U posljednjem slučaju 70% podataka će se koristiti kao trening set, 20% kao test set i 10% kao validacijski set podatka. Ukupan broj podataka koji se koristi u ovom radu je 651 datoteka. Kako bi se utvrdila efikasnost generalizacije mreže koristit će se formula za točnost na test skupu podataka:

$$Točnost = \frac{Ukupan\ broj\ primjera - Broj\ netočnih\ primjera}{Ukupan\ broj\ primjera} \tag{33}$$

Nakon provođenja 10 učenja konvolucijske neuronske mreže dobiveni su sljedeći rezultati:

Tablica 7. Rezultati učenja konvolucijske neuronske mreže

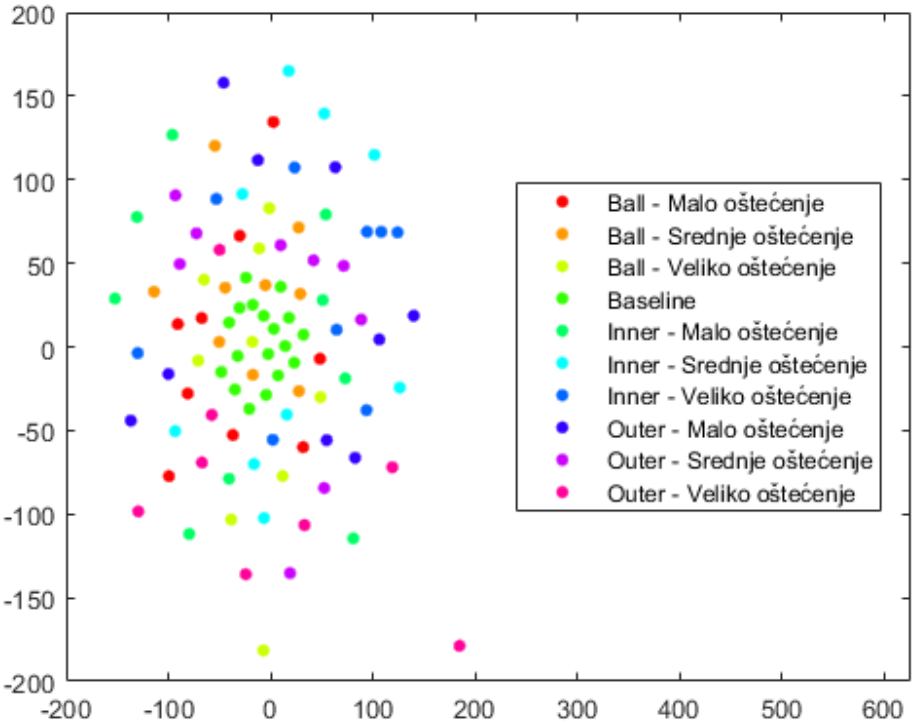
-	75% trening set 15% test set 10% validacijski set	60% trening set 30% test set 10% validacijski set	70% trening set 20% test set 10% validacijski set
Točnost	97,58 %	96,4 %	96,67 %
Standardna devijacija	1,51	1,77	1,64
Maksimalna točnost	100 %	99,49 %	99,24 %
Minimalna točnost	94,95 %	92,93 %	93,18 %

Iz tablice 7. vidljivo je da mreža ostvaruje najbolje rezultate procjene kvara u slučaju kada se koristilo 75% podataka kao trening set, 15% za test set i 10% za validaciju sa točnošću klasifikacije od 97,58%. Također, prema provedenom istraživanju, može se zaključiti kako generalno mreža bolje rezultate na test skupu postiže uz veći broj primjera na kojim je učena.

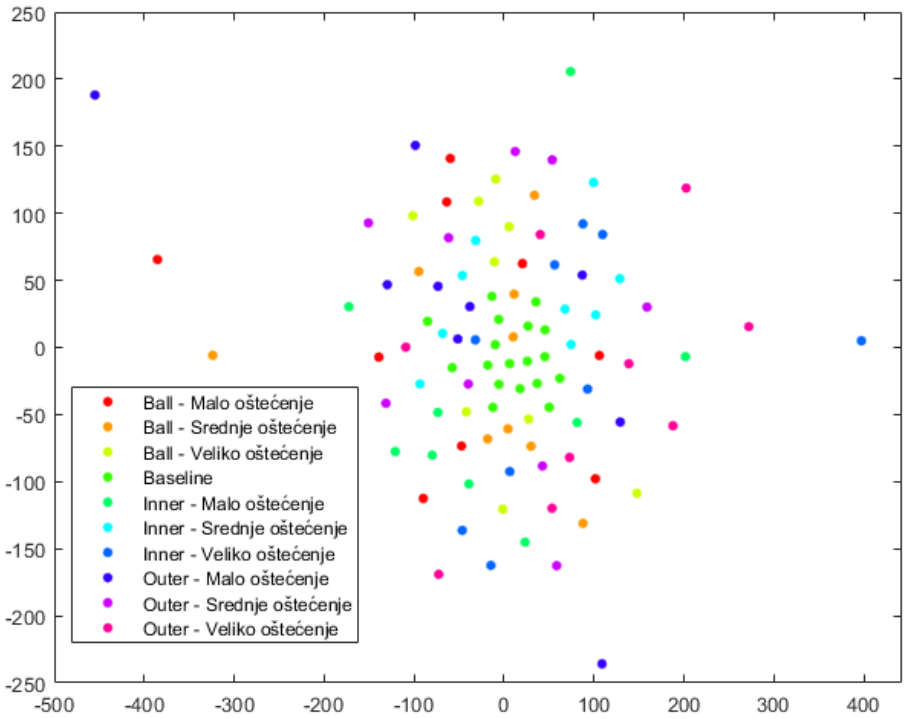
Kako bi se detaljnije opisao i vizualizirao postupak učenja značajku konvolucijskim neuronskim mrežama pokazat će se vizualni postupak grupiranja raznih značajki (u ovom slučaju kvarova ležajeva) tijekom prolaska kroz mrežu. Za to će se koristiti algoritam t-SNE (engl. *t-Distributed Stochastic Neighbor Embedding*) kojim će se izvlačiti vibracijske značajke. Ovim algoritmom vizualiziraju se podaci visokih dimenzija tako da se svakom od njih daje mjesto u dvodimenzionalnoj ili trodimenzionalnoj karti [50].

Na slici 35. prikazana je vizualizacija podataka u ulaznom sloj u konvolucijsku neuronsku mrežu. Vidljivo je da je teško razlikovati grupiranja u ovom sloju, uz malu iznimku podataka neoštećenih ležajeva (Baseline) koji su blago grupirani u sredini. Na slici 36. prikazana je

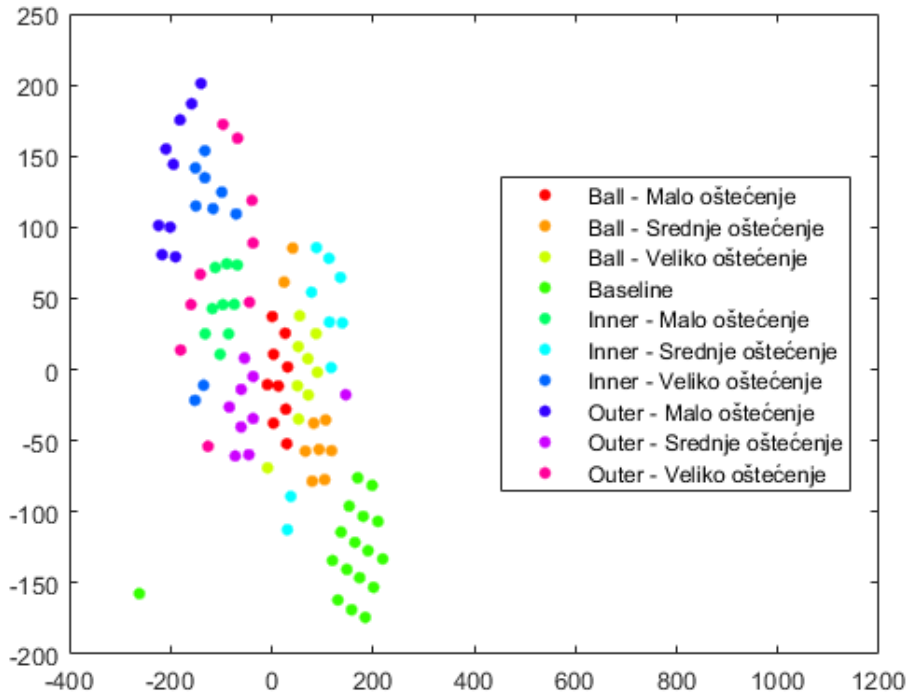
vizualizacija podataka u prvom konvolucijskom sloju gdje se još uvijek ne mogu razaznati grupe značajki, međutim sa slike 37. na kojoj je prikazan drugi konvolucijski sloj već se jasno vidi grupiranje različitih značajki. Na slici 38. prikazane su karakteristike u potpuno povezanom sloju, gdje se uspješno razlikuju 10 mogućih stanja ležajeva.



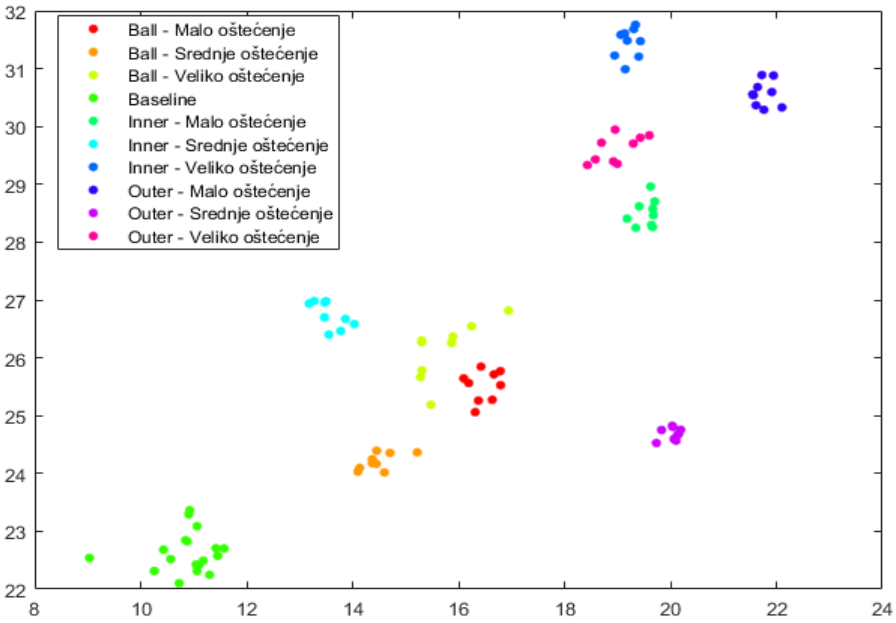
Slika 35. Vizualizacija značajki na ulaznom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma



Slika 36. Vizualizacija značajki u prvom konvolucijskom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma

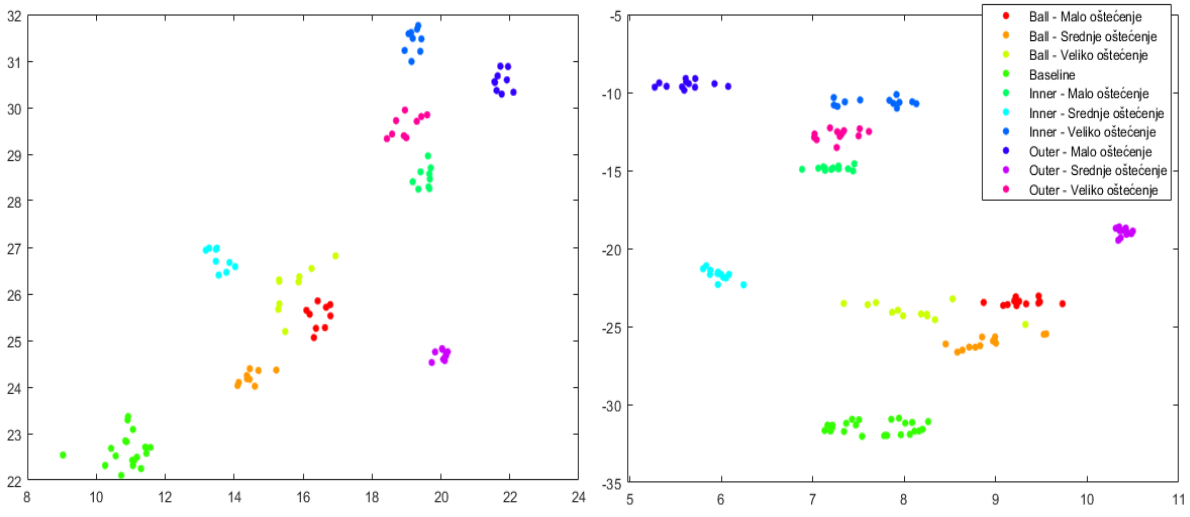


Slika 37. Vizualizacija značajki u drugom konvolucijskom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma



Slika 38. Vizualizacija značajki u potpuno povezanom sloju konvolucijske neuronske mreže uz pomoć t-SNE algoritma

Na slici 39. može se vidjeti razlika u vizualizaciji značajki u potpuno povezanom sloju konvolucijske mreže između najuspješnijeg slučaja (75% trening set, 15% test set, 10% validacijski set) i najneuspješnijeg slučaja (60% trening set, 20% test set, 10% validacijski set).



Slika 39. Vizualizacija značajki u potpuno povezanom sloju konvolucijske mreže uz pomoć t-SNE algoritma između najuspješnijeg slučaja (lijevo) i najneuspješnijeg slučaja (desno)

6. ZAKLJUČAK

Ležajevi su elementi strojeva koji služe za vođenje pokretnih strojnih dijelova te prenose vanjska opterećenja između tih dijelova i onih koji se nalaze u relativnom gibanju. Često su od presudnog značenja za integritet i vijek trajanja strojeva te njihovi kvarovi mogu imati ozbiljne posljedice. Ovim radom prikazane su mogućnosti primjene algoritma konvolucijskih neuronskih mreža u procjeni uzroka i intenziteta oštećenja na ležajevima čiji su podaci prikupljeni na sveučilištu Case Western Reserve. Podaci su prikupljani uz pomoć vibrodijagnostike, koja spada u jednu od najkorištenijih tehnika održavanja po stanju kod dijagnostike rotirajuće strojne opreme, a njome se mjere amplitude vibracija prilikom rotacije rotirajućih strojnih dijelova. Iako primjena metoda strojnih učenja u prediktivnom održavanju nad vibracijskim signalima ostvaruje impresivne rezultate, zbog vremenski zahtjevne implementacije te činjenice da zahtjeva znanje stručnjaka kako bi se odabrale adekvatne značajke koje bi se izvlačile i na temelju kojih bi se vršilo praćenje stanja sustava te donošenje odluka održavanja, primjena metoda dubokog učenja, tj. umjetnih neuronskih mreža poprima sve veći značaj. Njima se znatno reducira vrijeme izvlačenja značajki iz vibracijskih signala bez potrebe za specificiranjem značajki koje bi se trebale izvlačiti niti obrade ulaznih podataka. Prolaskom signala kroz višeslojnu strukturu umjetnih neuronskih mreža, mreža uspijeva samostalno naučiti njegove značajke. Primjena konvolucijskih neuronskih mreža uglavnom se veže uz vizijske sustave i obradu slike, međutim ovim radom se pokušala pokazati mogućnost njegove primjene u području raspoznavanja vibracijskih signala ležajeva sa raznim oštećenjima. Konvolucijskom neuronskom mrežom sastavljenom od dva uzastopna slijeda konvolucijskog, normalizacijskog i sažimajućeg sloja i završnog potpuno povezanog sloja ostvarena je točnost klasifikacije od 97,58 % na testnom skupu signala iz vremenske domene. Naučenu mrežu moguće je dalje koristi u procjeni postojanja oštećenja drugih ležajeva, uz pomoć njihovih vibracijskih signala. Konačno, u ovom radu prikazani su rezultati upotrebe algoritama konvolucijske neuronske mreže u procjeni uzroka i intenziteta kvarova ležaja, prema kojima je moguće tvrditi kako postoji potencijal primjene metoda dubokog učenja u održavanju po stanju.

LITERATURA

[1] European Federation of National Maintenance Society, Society Maintenance and Reliability Professionals: Global Maintenance and Reliability Indicators

[2] German Institute for Standardisation: DIN 31051 : 2012

[3] Lisjak D.: Bilješke s predavanja kolegija: Održavanje, ak.god. 2015./2016.

[4] Mobley R.K.: An introduction to predictive maintenance, Butterworth-Heinemann, SAD, 2002.

[5] Kothamasu R., Huang S.H., VerDuin W.H.: System health monitoring and prognostics – a review of current paradigms and practices. The International Journal of Advanced Manufacturing Technology, 2006., 28:1012–1024

[6] Čala I. : Prilog istraživanju elemenata organizacijske strukture održavanja strojarske opreme, Disertacija, Zagreb, 1998.

[7] Bralo K.: Analiza primjene suvremenih strategija održavanja u pomorstvu, Diplomski rad, Split, 2016.

[8] Sullivan G. P., Pugh R., Melendez A. P., Hunt W. D.: Operations & Maintenance Best Practices: A Guide to Achieving Operational Efficiency, US Department of Energy, SAD, 2010.

[9] Belak S., Čičin-Šain D.: Razvoj koncepta Terotehnologije. Pomorstvo, 2005., god. 19. 79-87

[10] Škalic B.: Primjena Ekspertnih sustava za održavanje suvremenih brodskih motora, Diplomski rad, Split, 2016.

[11] Mato Tudor: Samoodržavanje brodskih sustava. Pomorstvo, 2007.,2:31-38

[12] Kondić V., Horvat M., Maroević F.: Primjena dijagnostike kao osnove održavanja po stanju na primjeru motora osobnog automobila. Tehnički glasnik, 2013; 2:35-41

[13] Doko M.: Primjena metoda strojnog učenja u održavanju, Diplomski rad, Zagreb, 2018.

[14] Byington C.S., Kacprzynski G.J., Rnenzer M.J.: Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance. Proceedings, IEEE Aerospace Conference, 2002., 6:2815-2824

[15] Yan J.: Machinery Prognostics and Prognosis Oriented Maintenance Management, Wiley, Singapur, 2015.

[16] Randall R.B.: Vibration-based Condition Monitoring, Wiley, UK, 2011.

[17] Machine Learning, Andrew Ng, 04/22/2013-07/01/2013, https://www.coursehero.com/file/17680039/Machine-Learning/ - Pristup 29.12.2018.

[18] Kujundţić M.: Analiza modela predviđanja trendova kretanja cijena vrijednosnih papira, Diplomski rad, Rijeka, 2015.

[19] Kononenko I., Kukar M.: Machine learning and data mining: Introduction to Principles and Algorithms, Woodhead Publishing, UK, 2007.

[20] Goodfellow I., Bengio Y., Courville A.: Deep Learning, MIT Press; www.deeplearningbook.org

[21] https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d - Pristup 29.12.2018.

[22] Brownlee J., Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch, http://index-of.es/Varios-2/Master%20Machine%20Learning%20Algorithms.pdf – Pristup 29.12.2018.

[23] https://www.geeksforgeeks.org/clustering-in-machine-learning/ - Pristup 29.12.2018.

[24] https://www.forbes.com/sites/bernardmarr/2018/10/22/artificial-intelligence-whats-the-difference-between-deep-learning-and-reinforcement-learning/#2b92038b271e – Pristup 29.12.2018.

[25] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#1678fa6560ba – Pristup 29.12.2018.

[26] Dumančić S., Neuronske mreže, Diplomski rad, Osijek, 2014.

[27] Dalbelo Bašić B., Čupić M., Šnajder J.: Umjetne neuronske mreže, FER, Zagreb 2008.

[28] Bishop C.M.: Pattern Recognition and Machine Learning, Springer, SAD 2006.,

[29] https://www.automatika.rs/baza-znanja/neuralne-mreze/uvod-u-neuralne-mreze.html – Pristup 29.12.2018.

[30] https://skymind.ai/wiki/lstm#two – Pristup 29.12.2018.

[31] http://cs231n.github.io/neural-networks-1/ - Pristup 29.12.2018.

[32] Zadravec M.: Prepoznavanje teksta korištenjem konvolucijskih neuronskih mreža, Završni rad, Čakovec, 2017.

[33] Borko I.: Semantička segmentacija prirodnih scena dubokim neuronskim mrežama, Diplomski rad, Zagreb, 2015.

[34] http://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/ - Pristup 29.12.2018.

[35] https://towardsdatascience.com/gradient-descent-in-a-nutshell-eaf8c18212f0 – Pristup 29.12.2018.

[36] Kundid Vasić M.: Detekcija ljudi na zračnim slikama upotrebom konvolucijskih neuronskih mreža, Kvalifikacijski ispit, Split, 2017.

[37] https://brilliant.org/wiki/backpropagation – Pristup 29.12.2018.

[38] Michael Nielsen, Neural Networks and Deep Learning, 2015.

[39] https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/ - Pristup 29.12.2018.

[40] LeCun Y., Bottou L., Bengio Y., Haffner P.: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 1998.

[41] Krizhevsky A., Sutskever I., Hinton G.E.: ImageNet Classification with Deep Convolutional Neural Networks, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, 2012., 1097-1105

[42] Altenberger F., Lenz C: A Non-Technical Survey on Deep Convolutional Neural Network Architectures, ArXiv, 2018.

[43] https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/ - Pristup 29.12.2018.

[44] https://dlunizg.github.io/lab2/ - Pristup 29.12.2018.

[45] Janssens O., Van de Walle R., Loccufier M., Van Hoecke S.: Deep Learning for Infrared Thermal Image Based Machine Health Monitoring, Ieee-inst Electrical Electronics Engineers Inc, 2017., 23(1):151 - 159

[46] Khan S., Yairi T.: A review on the application of deep learning in system health management, Mechanical System and Signal Processing, 2017., 107: 241-265

[47] https://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website – Pristup 29.12.2018.

[48] Zhang W., Li C., Peng G., Chen Y., Zhang Z.: A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, Mechanical Systems and Signal Processing, 2018., 100: 439–453

[49] Tomislav Petković, Kratke upute za korištenje MATLAB-a, https://www.fer.unizg.hr/_download/repository/matlab_upute.pdf – Pristup: 29.12.2018.

[50] van der Maaten L., Hinton G.: Visualizing Data using t-SNE, Journal of Machine Learning Research, 2008; 9:2579-2605