**Spremljanje obrabe strojnih delov s strojnim sluhom**

Poročilo praktičnega projekta

Študent: Tilen Tinta

Mentor: izr. prof. dr. Simon Dobrišek

Predmet: Inteligentni avdio in govorni sistemi

Datum: 5.1.2025

Vsebina

[1 Uvod 2](#_Toc187105373)

[2 Fizikalno ozadje 2](#_Toc187105374)

[3 Eksperiment 2](#_Toc187105375)

[3.1. Deljenje vzorcev 2](#_Toc187105376)

[3.2. Večanje zbirke 2](#_Toc187105377)

[3.3. Pridobivanje značilk 3](#_Toc187105378)

[3.3.1. Spektralni centroid 3](#_Toc187105379)

[3.3.2. Spektralna širina 3](#_Toc187105380)

[3.3.3. Nizkofrekvenčna energija 4](#_Toc187105381)

[3.3.4. Visokofrekvenčna energija 4](#_Toc187105382)

[3.3.5. Povprečna amplituda 4](#_Toc187105383)

[3.3.6. Varianca 4](#_Toc187105384)

[3.3.7. Asimetrija / Skewness 4](#_Toc187105385)

[3.3.8. Kurtoza / Kurtosis 4](#_Toc187105386)

[3.3.9. Število vrhov 4](#_Toc187105387)

[3.4. Učenje razpoznavalnika 4](#_Toc187105388)

[3.4.1. Algoritem podpornih vektorjev – SVM 5](#_Toc187105389)

[3.5. Validacija učenja 5](#_Toc187105390)

[4. Izboljšave 6](#_Toc187105391)

[5. Zaključek 6](#_Toc187105392)

[6. Reference 7](#_Toc187105393)

**Ključne besede: zvočni posnetki, kavitacija, značilke, klasifikacija, SVM**

# Uvod

Praktični del seminarja nosi enak naslov kot študijski del. Ta dopolnjuje teoretično ozadje s primerom implementacije razpoznavalnika obrabe strojnih delov.

V tem seminarju smo se ukvarjali z razvrščanjem industrijskih zvokov, natančneje zvokov, ki jih povzroča tekočina ob pretoku skozi ventile. Na teh se je izračunalo značilke in naučilo razpoznavalnik. Ta ločuje zvoke v posamezne razrede, ki nam nato povejo ali so zvoki delovanja ustrezni ali ne. Če ta ne ustreza pravilnemu delovanju lahko posledično škodi strojnemu delu. Za razvrščanje je bil uporabljen algoritem SVM, ki zaradi svoje računske zahtevnosti za specifičen primer omogoča takemu sistemu delovanje na manj zmogljivi napravi.

# Fizikalno ozadje

Kot omenjeno se v tem seminarju ukvarjamo z zvokom pretoka tekočine skozi ventile. Ta nam omogoča pridobitev informacij o stanju ter pravilnosti pretoka iz notranjosti ventila katera v večini primerov ni vidna. Zvok povzroča tudi vibracije, ki jih lahko ravno tako merimo z namenskimi vibracijskimi senzorji.

Pojav, ki nas zanima v tem primeru je kavitacija. Ta se pojavi ko lokalen tlak v tekočini pade pod tlak nasičene pare tekočine. Pojavi se ob hitro vrtečih se delih kot so lopatice in propelerji ali ostrejših robovi. Enako se lahko ta pojavi ko mehurčki zraka ali nekega plina pridejo v območje z višjim tlakom.

V osnovi poznamo tri vrste kavitacije:

* Hidridinamična kavitacija (pojavi se ob propelerjih, črpalkah, lopaticah… zaradi prej omenjenih vzrokov)
* Akustična kavitacija (pojavi se zaradi ultrazvočnih valov z visoko amplitudo)
* Vibracijska kavitacija (povzročajo jo vibracije, ki ustvarjajo področja z nizkim tlakom)

V napravah ta seveda ni zaželena saj povzroča erozijo materiala, izgubo učinkovitosti v črpalkah ter seveda hrup in vibracije, ki nas v tem seminarju zanimajo. Pojav želimo kar se da zgodaj odkriti in odpraviti saj v nasprotnem primeru lahko vodi do uničenja strojnega dela.

# Eksperiment

## Deljenje vzorcev

Podatkovna zbirka zvokov uporabljenih v tem seminarju z naslovom »Sounds of valves in heating systems for classification and condition monitoring« je bila ustvarjena na strojni fakulteti univerze v Ljubljani. Vsebuje 427 posnetkov dolžine 3 sekunde, njihove oznake pa je mogoče najti v priloženi datoteki. Ker smo za učenje in testiranje želeli urejen nabor posnetkov so bili v prvem koraku na osnovi dokumenta z opisi razdeljeni v ločene datoteke.

V programskem jeziku *Python* je bil napisan program, ki iz datoteke z oznakami pridobi ime posnetka in razred v katerega spada. Na osnovi tega se posnetek ustrezno razvrsti.

Ta korak je bil potreben za preglednejšo obdelavo posnetkov in dodeljevanje oznak.

## Večanje zbirke

Po urejanju posnetkov je bilo takoj možno opaziti, da je posnetkov napak oz. kavitacije veliko manj od pravilnega delovanja. To potrjuje idejo, ki jo opisuje poglavje o transfernem učenju iz študijskega dela seminarja. Ker v tem primeru te možnosti ni bilo smo v ta namen umetno povečali nabor posnetkov.

Vsak posnetek je imel na začetku dolžino 3 sekund. Ker se zvok ponavlja oz. je ta čez celoten posnetek konstanten smo lahko te brez težav razrezali na krajše dele. Za to smo uporabili oknjenje. Čez posnetek smo »zapeljeli« okno dolžine 500 milisekund. Pri vsakem naslednjem premiku smo

upoštevali še 100ms prekrivanja. To nam je iz vsakega posnetka dalo osem krajših.

|  |  |  |
| --- | --- | --- |
| Razred | Št. posnetkov | Novo št. posnetkov |
| Cavitation | 102 | 816 |
| Flow | 279 | 2232 |
| Rattling | 23 | 184 |
| Whistling | 23 | 184 |

Tabela 3‑1: Bogatenje množice posnetkov

S tem postopkom smo pridobili veliko večjo množico posnetkov primernih za učenje in test razpoznavalnika. Posnetki so bili še vedno »čisti« in niso vsebovali namenoma vnesenega šuma. V primeru, da bi se izkazalo, da je takšna zbirka še vedno slaba bi bila to ena od možnosti za augmentacijo.

## Pridobivanje značilk

Za učenje razpoznavalnika sami posnetki niso najbolj primerni ampak je iz njih potrebno pridobiti značilke, ki jih kar se, da dobro predstavljajo.

V ta namen so bile iz vsakega zvoka izpeljane značilke, ki so bile shranjene v formatu csv za kasnejšo uporabo. Ker so te odvisne od domene v kateri obravnavamo posnetek smo za to po en primer posnetka iz vsakega razreda odprli z orodjem Audacity. Po pregledu spektrograma je je bilo mogoče opaziti razlike.

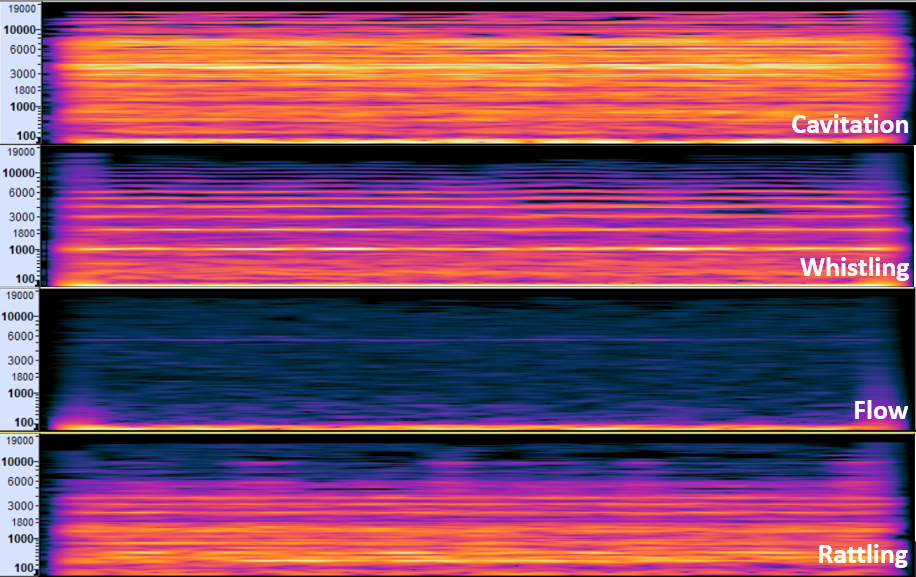


Tabela 3‑2: Spektrogrami posnetkov iz posameznega razreda

Ker so si nekateri od posnetkov med seboj po spektrogramih precej podobni smo za njihovo predstavitev uporabili še Audacity-evo orodje za hitri FFT. To je pokazalo lepše razlike med

razredi zato smo se odločili vse značilke izpeljati v frekvenčnem prostoru.

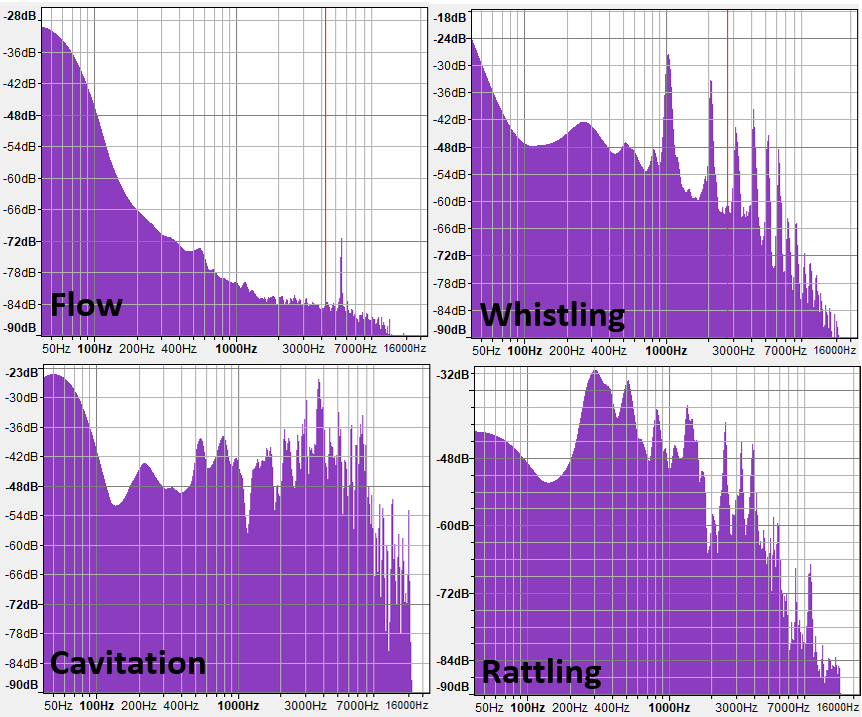


Tabela 3‑3: FFT sprektrogrami posnetkov po razredih

Na osnovi teh odločitev se je vsak posnetek najprej preverilo, da je ta v mono formatu in nato tega s pomočjo FFT-ja pretvorilo v frekvenčno domeno. Po normalizaciji se je na osnovi frekvenčnega spektra izračunalo naslednje značilke.

## Spektralni centroid

Predstavlja težišče spektra zvoka in opisuje, kje v spektru je koncentrirana energija. Pogosto je povezan s tem, kako "svetel" ali "oster" je zvok. Vrednost je podana kot frekvenca. Izračuna se ga po naslednji formuli.

(1)

Kjer je:

* : frekvenca
* : amplituda spektra pri frekvenci

## Spektralna širina

Je mera za širino spektra okoli spektralnega centroida. Pove nam kako širok oz. razpršen je spekter. Ravno tako je tudi ta mera podana kot frekvenca. Izračuna se jo po naslednji formuli.

(2)

Kjer je:

* *SC*: spektralni centroid
* : frekvenca
* : amplituda spektra pri frekvenci

## Nizkofrekvenčna energija

Opisuje delež energije, ki pripada nižjim frekvenčnim komponentam zvoka. Uporablja se za ločevanje globokih zvokov. Izračuna se po naslednji formuli.

(3)

Kjer je:

* : prag za nizke frekvence
* : frekvenca
* : amplituda spektra pri frekvenci

## Visokofrekvenčna energija

Podaja delež energije, ki pripada višjim frekvenčnim komponentam zvoka. Povezan je z ostrino ali visokimi toni. Izračuna se po:

(4)

Kjer je:

* : prag za nizke frekvence
* : frekvenca
* : amplituda spektra pri frekvenci

## Povprečna amplituda

Povprečna amplituda signala v časovni domeni predstavlja povprečno intenziteto zvoka. Izračuna se jo kot:

(5)

Kjer je:

* : trenutna amplituda
* : število vzorcev v signalu

## Varianca

To je mera za razpršenost amplitud signala. Predstavlja, kako konstanten ali spremenljiv je

zvok, kar je v naši aplikaciji zelo pomembno. Izračuna se preko naslednje formule.

(6)

Kjer je:

* : trenutna amplituda
* : povprečna vrednost amplitud (mean)

## Asimetrija / Skewness

Je mera za asimetrijo distribucije amplitud signala. Pokaže ali se amplitude bolj nagibajo proti pozitivnim ali negativnim vrednostim

(7)

Kjer je:

* : trenutna amplituda
* : povprečna vrednost amplitud (mean)
* : standardni odklon

## Kurtoza / Kurtosis

Poznana tudi kot mera za "špičatost" distribucije amplitud. Visoka kurtoza pomeni, da je večina energije skoncentrirana okoli povprečja z redkimi visokimi odkloni.

(7)

Kjer je:

* : trenutna amplituda
* : povprečna vrednost amplitud (mean)
* : standardni odklon

## Število vrhov

Kot že samo ime pove je to število lokalnih vrhov v amplitudi signala. Predstavlja, kako dinamičen je signal. Primeren je ko iščemo na primer vibracije.

## Učenje razpoznavalnika

Iz prodobljenih značilk se je tem pred shranjevanjem dodala še oznaka razreda kamor posamezen posnetek spada ter ime

posnetka na katerem so bile te pridobljene. Slednje ni uporabno za učenje in test razpoznavalnika, pomemben pa je razred. Kljub naprednim metodam, ki temeljijo na umetni inteligenci smo se pri tej aplikaciji odločili za »enostavnejšo« metodo. Uporabljen je bil algoritem podpornih vektorjev oz. SVM (angl. support vector machine). Za to metodo smo se odločili na podlagi poslušanja posnetkov ter FFT slik, ki kažejo na zelo izrazite frekvenčne vrhe oz. jasno slišne razlike med razredi. Na osnovi tega smo sklepali, da delitev v razrede ne bi smela biti prezahtevna in je uporaba nevronskih mrež nepotrebna.

## Algoritem podpornih vektorjev – SVM

Osnovna ideja SVM-jev je iskanje ločilne meje med razredi, ki ima največjo razdaljo do vzorcev. Podobno idejo imajo tudi nevronske mreže kot je klasičen primer perceptrona le, da je ta v celoti matematičen in ponovljiv z istim rezultatom.

Za algoritem je bila seveda uporabljena knjižnica *sklearn* v programskem jeziku *Python.* Program je uvozil vse csv datoteke, ki vsebujejo značilke posnetkov, jih združil in iz nabora izbrisal imena posnetkov iz katerih so bile značilke pridobljene. Ta nabor se je nato še posebej razdelil na značilke in oznake razredov. Na značilkah je bila nato opravljena normalizacija in dodatno še binarizacija, ki je potrebna za izračun ROC krivulje. Celoten nabor se je razdelil na dva dela v razmerju 80:20 za učno in testno množico. Urejeni podatki so bili tako primerni za učenje SVM-ja.

SVM se je učilo s funkcijama *OneVsRestClassifier* in *SVC*. Druga je klasičen binarni klasifikator, kar v našem primeru ne zadostuje. V ta namen se uporabi prvo funkcijo, ki omogoča več dimenzionalno klasifikacijo. Ravno tako je bilo za učenje uporabljeno *rbf* jedro (angl. Radial Biasis Function), ki je nelinearno. Ostali parametri kot so *C, gamma,*

*probability* in *random\_state* so ostali nastavljeni na privzete vrednosti.

Parametri:

* **C**: uravnava kompromis med točnostjo klasifikacije na učnem vzorcu in širino meje največje razdalje. Manjša vrednost poda več napak na učnih vzorcih, a naredi bolj splošen model, večja vrednost pa naredi manj napak na učnem vzorcu a lahko pa vodi do prenaučenosti (overfitting).
* **Gamma**: določa, kako daleč vpliva posamezna učna točka. Vrednost *scale,* ki je bila izbrana se prilagaja samodejno glede na podatke.
* **Probability**: vrača vrednosti pripadnosti posameznemu razredu. Uporablja se metoda Plattova kalibracija.
* **Random\_state**: nastavi in ohrani začetno stanje generatorja naključnih števil. To zagotavlja ponovljivost rezultatov, saj bo tako model z istimi podatki in parametri vedno dal enak rezultat. To je tudi ena od glavnih prednosti SVM-ja

## Validacija učenja

Po zaključenem učenju, ki je v našem primeru trajalo na srednje zmogljivem računalniku le nekaj sekund smo dobili obetajoče rezultate. Ti so:

***Accuracy: 0.9532163742690059***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | F1-score | support |
| Cavitation | 0.98 | 0.99 | 0.99 | 189 |
| Flow | 0.94 | 1.00 | 0.97 | 426 |
| Rattling | 1.00 | 0.19 | 0.32 | 0.32 |
| Whistling | 1.00 | 0.86 | 0.93 | 37 |
| Accuracy |  |  | 0.95 | 684 |
| Marco avg. | 0.98 | 0.76 | 0.80 | 684 |
| Weighted avg. | 0.96 | 0.95 | 0.94 | 684 |

Tabela 3‑4: Poročilo SVM klasifikacije

Vrednosti v zgornji tabeli so:

* **Accuracy / točnost**
* **Precision / natančnost**: je delež primerov označenih kot določen razred in dejansko njemu tudi propadajo
* **Recal**: je delež primerov, ki pripadajo določenemu razredu in je model pravilno prepoznal
* **F1-score**: je harmonična sredina natančnosti in priklica. Uporablja se za uravnoteženo oceno modela, še posebej, če sta natančnost in priklic različna
* **Support**: število primerov v vsakem razredu
* **Macro avg**: je povprečje vrednosti čez vse razrede, pri čemer so vsi razredi enako pomembni ne glede na število vzorcev.
* **Weightd avg**: je povprečje vrednosti vteženo glede na število vzorcev (support) v razredu. Razredi z več vzorci imajo večji vpliv na to vrednost.

Natančnost je lahko podamo tudi kot ROC krivuljo za vsak razred posebej.

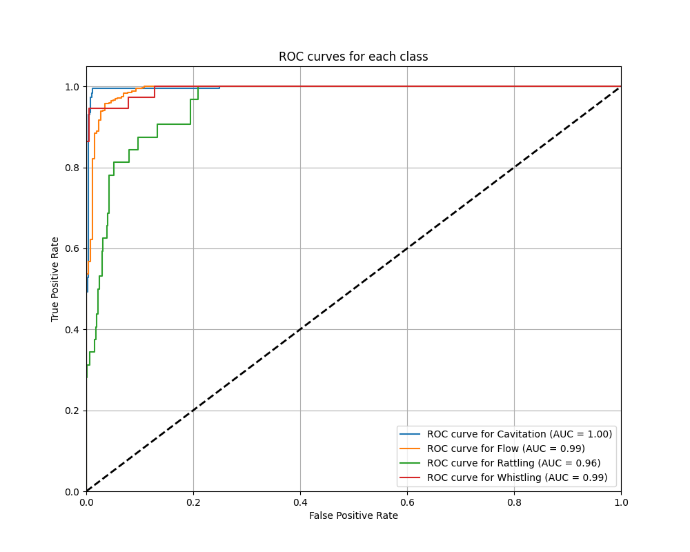


Tabela 3‑5: ROC krivulja za vsak razred

# Izboljšave

Trenutni rezultati kažejo na zelo dobro delovanje trenutnega klasifikatorja, ima pa ta še možnosti izboljšanja.

Prva možnost bi bila spreminjanje oz. fino nastavljanje parametrov uporabljenih za učenje SVM-ja. S tem so mišljeni tako C, gamma kot random\_state… v primeru uporabe tega

algoritma za nelinearne meje. Lahko bi se spremenilo tudi algoritem za računanje mej.

Kot drugo možnost izboljšave bi definitivno lahko za klasifikacijo uporabili nevronsko mrežo. Klasičen CNN ali pa tudi osnoven perceptron bi bila primerna kandidata za takšno klasifikacijo z malo značilkami.

Če za izboljšanje razpoznavanja ohranimo trenutni algoritem, bi to lahko izboljšali z več učnimi podatki oz. dodatno augmentacijo obstoječih. To bi lahko storili s dodajanjem šuma na posnetke, vsiljevanjem motenj na neki frekvenci, dodajanjem enosmerne komponente zvoku…

# Zaključek

V tej seminarski nalogi smo raziskali praktičen primer implementacije sistema za razpoznavanje obrabe oz. detektiranje napake, ki bi povzročila obrabo strojnih delov. Izvedli smo vse korake, ki sledijo od pridobitve vzorcev naprej. To so urejanje zbirke na osnovi oznak, bogatenje učne in testne množice, pridobivanje značilk ter ustrezno shranjevanje le teh, učenje SVM-ja in evaluacija rezultatov.

Rezultati projekta so v trenutni konfiguraciji dobri a imajo še vseeno možnosti izboljšanja. Kljub naprednim metodam, ki so v zadnjih časih vedno bolj popularne (umetna inteligenca) lahko jasno vidimo, da so tudi klasične matematične metode ob pravih pogojih ravno tako dobre.

Velika prednost, ki jo vidimo v tej specifični aplikaciji je seveda računska zahtevnost na strani računalniške opreme. Celotna obdelava podatkov in učenje je bila izvedena brez grafične kartice in to zelo hitro. To nakazuje, da je takšna metoda zelo primerna za edge-computing, katerega opisuje tudi študijski del seminarja.

# Reference

1. Drsnice s predavanj pri predmetu: Inteligentni avdio in govorni sistemi, prof. dr. Simon Dobrišek
2. Drsnice s predavanj pri predmetu: Razpoznavanje vzorcev, prof. dr. Simon Dobrišek
3. Luščenje značilk za nadaljno učenje: CNNs for Audio Classification [https://towardsdatascience.com/cnns-for-audio-classification-6244954665ab]
4. Referenčna navodila za uporabo sklearn: 1.4. Support Vector Machines [https://scikit-dlearn.org/1.5/modules/svm.html]
5. Primer uporabe in opis SVM-ja: Support Vector Machine (SVM) Algorithm

[https://www.geeksforgeeks.org/support-vector-machine-algorithm/]

1. SVM vodič: SVM Classifier Tutorial [https://www.kaggle.com/code/prashant111/svm-classifier-tutorial]
2. Potočnik, Primož; Vodopivec, Lučka; Susič, Egon (2022), “Sounds of valves in heating systems for classification and condition monitoring”, Mendeley Data, V2, doi: 10.17632/y6fkrybb32.2