

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Saša Saftić

**Prototip sistema za zajem in  
klasifikacijo dojenčkovega joka**

MAGISTRSKO DELO

MAGISTRSKI PROGRAM DRUGE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Mojca Ciglarič

SOMENTOR: prof. dr. Blaž Zupan

Ljubljana, 2016



AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

©2016 SAŠA SAFTIĆ



## ZAHVALA

*Zahvaljujem se mentorici in somentorju za ves porabljen čas, preglede in komentarje pri izdelavi te magistske naloge. Zahvaljujem se tudi staršema, ki sta mi omogočila, da lahko študiram in sta me tekom študija podpirala tako moralno kot finančno. Zahvaljujem se tudi Teotu, ki me je v trenutkih pomanjkanja motivacije spodbudil k ponovnemu delu.*

*Saša Saftić, 2016*



# Contents

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled sorodnih del</b>	<b>3</b>
2.1	Uvod . . . . .	3
2.2	Klasifikacije v razrede patologij . . . . .	3
2.3	Rezultati: Pregled sorodnih del . . . . .	4
<b>3</b>	<b>Zajem podatkov</b>	<b>9</b>
3.1	Baze podatkov . . . . .	9
3.2	Objavljeni video posnetki . . . . .	10
3.3	Zajem podatkov z aplikacijo Android . . . . .	11
<b>4</b>	<b>Hramba in obdelava podatkov v oblaku</b>	<b>15</b>
4.1	Kaj je oblak . . . . .	15
4.2	Virtualizacija . . . . .	16
4.3	Ključne značilnosti . . . . .	17
4.4	Modeli storitve . . . . .	18
4.5	Modeli namestitve . . . . .	18
4.6	OpenStack . . . . .	19
4.7	Django REST API . . . . .	19
4.8	Django . . . . .	23

## CONTENTS

4.9	Uporabljena infrastruktura . . . . .	23
<b>5</b>	<b>Napovedni model razloga za jok</b>	<b>25</b>
5.1	Zgradba modela joka . . . . .	25
5.2	Pridobivanje značilk . . . . .	26
5.3	Preveliko prilagajanje modela . . . . .	30
5.4	Opis metod (večrazrednih) klasifikacij . . . . .	30
5.5	Orange . . . . .	37
<b>6</b>	<b>Metodologija in rezultati</b>	<b>39</b>
6.1	Obdelava posnetkov . . . . .	39
6.2	Pridobitev značilk . . . . .	40
6.3	Priprava okolja na oblaku za učenje napovednih modelov . . .	40
6.4	Binarna klasifikacija . . . . .	41
6.5	Večrazredna klasifikacija . . . . .	44
6.6	Rekurenčne nevronske mreže . . . . .	45
6.7	Primerjava po starosti . . . . .	47
<b>7</b>	<b>Diskusija in nadaljnje delo</b>	<b>49</b>
<b>A</b>	<b>Jok</b>	<b>53</b>
A.1	Lastnosti zvočnega signala pri joku . . . . .	54
A.2	Individualne in skupinske razlike v proizvajanju zvokov med jokom . . . . .	58
<b>B</b>	<b>Primerjava mer uspešnosti za večrazredno klasifikacijo</b>	<b>61</b>
<b>C</b>	<b>Podroben opis posameznih posnetkov joka iz Youtube-a</b>	<b>63</b>
<b>D</b>	<b>Primer nevronske mreže</b>	<b>69</b>



# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>BFBE</b>	Bark-filter band energy	bark zaporedje časovnih filtrov
<b>CA</b>	classification accuracy	klasifikacijska natančnost
<b>CCA</b>	Canonical correlation analysis	analiza kanonične korelacije
<b>CNN</b>	convoluted neural network	konvolucijske nevronske mreže
<b>f</b>	f requency	frekvenca
<b>f(0)</b>	fundamental frequency	osnovna frekvenca
<b>F1</b>	first formant	prva formantna frekvenca
<b>F2</b>	second formant	druga formantna frekvenca
<b>FFNN</b>	feed-forward neural network	Usmerjene nevronske mreže
<b>FFT</b>	Fast Fourier Transform	Hitra Fourierova transformacija
<b>HMM</b>	Hidden Markov Models	skriti Markov model
<b>k-NN</b>	k-nearest neighbours	k-najbližjih sosedov
<b>LFCC</b>	Linear Frequency Cepstral Coefficients	linearno-frekvenčni kepralni koeficienti
<b>LPC</b>	linear prediction coeficients	linearno-napovedni koeficienti
<b>LPCC</b>	Linear Prediction Cepstral Coefficients	linearno-napovedni kepralni koeficienti
<b>LR</b>	Logistic regression	logistična regresija
<b>MFBE</b>	Mel-filter band energy	energija Melovega zaporedje pasovnih filtrov
<b>PCA</b>	principle component analysis	metoda glavnih komponent
<b>PS</b>	Power spectrum	močnostni spekter
<b>RBFN</b>	Radial Basis Function Network	
<b>RNN</b>	recurrent neural network	rekurenčne nevronske mreže
<b>SVM</b>	support vector machine	metoda podpornih vektojev



# Povzetek

**Naslov:** Prototip sistema za zajem in klasifikacijo dojenčkovega joka

V tem delu se ukvarjamo z večrazredno klasifikacijo dojenčkovega joka in raziskovanjem povezave med starostjo dojenčka in natančnostjo klasifikacije. Prav tako se ukvarjamo z varnim shranjevanjem in obdelavo podatkov v oblaku. Primerjali smo več standarnih modelov za večrazredno klasifikacijo in klasifikacijo z rekurenčnimi nevronskimi mrežami. Natančnost klasifikacije smo preverili na podatkih različno starih otrok. Za shranjevanje in obdelavo podatkov smo uporabili Rest Django API in odptokodno oblačno platformo OpenStack. Rezultati dela kažejo, da lahko s pomočjo večrazredne klasifikacije razlikujemo med razredi joka. Povezave med starostjo otroka in natančnostjo klasifikacije nismo opazili. Django Rest API in OpenStack platformi sta se izkazali za dobro orodje za shranjevanje in obdelavo podatkov v oblaku.

## Ključne besede

*dojenčkov jok, zajem podatkov, analiza zvoka, odkrivanje značilnk, klasifikacija, obdelava podatkov v oblaku, oblačna varnost*



# Abstract

**Title:** System for audio capture and classification of baby cry samples

We explore multiclass classification of infants' crying and the relation between the age of the infant and the accuracy of classification. Additionally we explore secure cloud storage and cloud data processing. We compare several state-of-the-art multiclass classification models with recurrent neural networks. Classification accuracy was obtained on data from infants of various ages. For data storage and processing we used the Django Rest API and the opensource cloud platform OpenStack. Multiclass classification models successfully differentiated between different classes of crying, but no age effect has been found. We have demonstrated the aptness of the Django Rest API and OpenStack platform for data storing and processing in the cloud.

## Keywords

*infant cry, data acquisition, audio analysis, feature extraction, classification, cloud processing, cloud security*



# Poglavje 1

## Uvod

Jok je način komunikacije dojenčka, ki ga uporablja poleg obrazne mimike in gibanja telesa. Za starše, ki za otroka skrbijo, je pomembno, da razumejo, kaj dojenček potrebuje oziroma kako se počuti. Za zdravnika, ki dojenčka želi pregledati, pa je pomembno, kaj mu dojenček sporoča. Zaradi nerazumevanja dojenčkovega joka je komunikacija med dojenčkom in odraslimi pomanjkljiva. Številne raziskave potrjujejo, da lahko z obdelavo podatkov klasificiramo otroški jok. Večina raziskav se osredotoča na razlikovanje med zdravim in bolnim dojenčkom, manj pa na klasifikacijo dojenčkovega počutja [1]. Pri pregledu literature nismo našli študije, ki bi raziskovala povezavo med starostjo dojenčka in uspešnostjo klasifikacijskih algoritmov. V magistrski nalogi raziskujemo ali višja starost otroka in s tem večja razvitost telesa in možganov vpliva na klasifikacijsko uspešnost dojenčkovega joka in ali lahko naučeni model uspešno uporabimo za klasifikacijo novih posnetkov dojenčkovega joka.

V poglavju 2 opravimo sistematičen pregled literature in iz njega izpeljemo vodila za raziskavo. V naslednjih dveh poglavjih opišemo pridobivanje posnetkov, ki jih uporabimo za učno in testno množico. V poglavju 4 opišemo uporabljene platforme, ki so nam omogočile shranjevanje in obdelavo podatkov v oblaku. V poglavju, ki sledi opišemo zgradbo modela, pridobivanje značilnk in posamezne modele, ki smo jih uporabljali pri klasifikaciji. V

poglavju 6 sistematično opišemo metodologijo našega dela in pridobljene rezultate na vsakem koraku metodologije. Zaključimo z diskusijo o rezultatih, opišemo nadaljne delo in podamo sklepne ugotovitve.



# Poglavje 2

## Pregled sorodnih del

### 2.1 Uvod

Raziskovanje dojenčkovega joka se je v preteklih dveh desetletjih močno usmerilo v avtomatične klasifikacije joka, ki na podlagi posnetka joka avtomatično uvrstijo jok v ciljni razred. V splošnem poznamo dve smeri raziskav: klasifikacije patoloških jokov glede na bolezensko stanje in zdravih jokov v razloge za jok.

### 2.2 Klasifikacije v razrede patologij

Klasifikacijski modeli dojenčkovih patoloških jokov so dosegli že izredno uspešne napovedi (tj. 90% natančnost uvrščanja in več). Avtomatično lahko ločijo med zdravim in bolnim dojenčkom ali uvrstijo jok v razrede bolezenskih stanj. Danes lahko iz joka napovemo, če ima otrok slušno okvaro [2, 3], hipotiroidizem ali moteno delovanje ščitnice [4], asfiksijo ob rojstvu [5], rizik nevroloških motenj [6], in številne druge bolezni. Pri pregledu literature smo se omejili na klasifikacije v nepatološke razrede, saj že obstajajo številni odlični pregledi klasifikacij patologij iz dojenčkovih jokov [7, 1]. Joki dojenčkov s patologijo se akustično drastično razlikujejo od jokov zdravih dojenčkov v osnovnih in formantnih frekvencah, močnostnem spektru in drugih prvinah [8]. Velike

razlike med zdravim in patološkim jokom namigujejo, da je potrebno pri klasifikaciji zdravih jokov uporabiti drugačne prijeme, saj zdravi joki nimajo tako edinstvenih akustičnih prvin. Slednje deloma razloži presenetljivo visoka natančnost klasifikacije, saj je na primer jok gluhega dojenčka tudi več tisoč Hz višji od joka zdravega dojenčka [1].

## 2.3 Rezultati: Pregled sorodnih del

Tabela 2.1 povzema vse sistematično pridobljene vire v kronološkem vrstnem redu njihovih objav. Za vsak vir povzema avtorje, tarčne razrede za klasifikacijo, izbrane značilke, metodo klasifikacije ter mero in nivo uspešnosti klasifikacije.

**Tabela 2.1:** Pregled sorodnih del

Avtorji	Ciljni razred	Pridobivanje značilk	Metoda klasifikacije	Uspešnost klasifikacije	Mera uspešnosti
Petroni et al. (1995) [9]	jeza strah bolečina	MFCC MFBE BFBE	FFNN	84%	CA
			RNN	70%	
			TDNN	63%	
			CCNN	59%	
Petroni et al. (1995) [10]	jeza strah bolečina	MFCC	FFNN	69.2%	CA
			RNN	64.6%	
			TDNN	61.0%	
Orozco Garcia & Reyes Garcia (2002) [11]	lakota bolečina patologija	LPC	SCGNN	70.58%	CA
				79.16%	
				50%	
				popv. 73.33%	
Baeck & Souza (2003) [12]	bolečina ne bolečina	$f(0) + F1 + f$	Bayesov klasifikator	75%	CA
Barajas & Reyes (2005) [13]	bolečina lakota	LPC + MFCC	GFNN	95.24%	CA
Barajas-Montiel & Reyes-Garcia (2005) [14]	lakota ne lakota	MFCC	AdaBoost + FFNN	87.61%	CA
				bolečina ne bolečina	
	lakota ne lakota		AdaBoost + SVM	76.41%	
				bolečina ne bolečina	
Barajas-Montiel & Reyes-García (2006) [15]	bolečina lakota ne bolečina ne lakota	MFCC	F-SVM	98%	CA
Mima & Arakawa (2006) [16]	lakota zaspanost neudobje (mokra planica/lega telesa)	PS	RBS	85%	CA
Pal et al. (2006) [17]	bolečina lakota strah	$f(0) + F1 + F2$	k-NN	91%	CA
				72%	
				71%	

	žalost			79%	
	jeza			58%	
				povp. 74.2%	
Abdulazis & Syed Ahmad (2010a) [18]	bolečina ne bolečina (lakota/jeza)	MFCC LPCC	SCGNN	76.20%	CA
Abdulazis & Syed Ahmad (2010b) [19]	bolečina ne bolečina	MFCC LPCC	HMM	71.8%-92.3%	CA
Kuo (2010) [20]	lakota mokre planice iskanje pozornosti	LPCC	LVQ	70-74%	CA
Rosales-Pérez, et al. (2011) [21]	lakota bolečina	MFCC + PCA	GFRNN	96% 98.4%	CA AUC
Honda et al. (2012) [22]	zaspanost + odpor jeza + žalost + lakota	PS + MFCC + f	HMM	74.7%	?
Molaezadeh et al. (2012) [23]	lakota bolečina	MFCC	T2 FPM SVM LR	97.8% 95.7% 93.5%	CA
Rosales-Pérez et al. (2012) [24]	lakota bolečina	MFCC	GSFM	97.96% 98.90%	CA AUC
Vempada et al. (2012) [25]	mokra planica lakota bolečina	MFCC STE MFCC + STE	SVM	80.56%	CA
Liu et al. (2013) [26]	lakota mokra planica iskanje pozornosti	LPCC MFCC BFCC	? (najverjetneje Bayesov klasifikator)	80%	?
Singh et al. (2013) [27]	lakota bolečina mokra planica	MFCC + EIC	GMM	79%	CA
Srijiranon & Eiamkanitchat (2014) [28]	napihnjenost neudobje lakota zaspanost	MFCC LPCC PLP RASTA	MLP + fuzzy logic + k-NN	86.30%	CA
Bano & Ravi Kumar (2015) [29]	lakota bolečina pika/sčipa bolečina ušes/trebuha zaspanost neudobje	MFCC	k-NN	90%	CA
Ntalampiras & Potamitis (2015) [30]	lakota neudobje (mokra planica) napihnjenost bolečina zaspanost	MFCC TEO CCA	SVM	91.70%	CA
Varsharani & Sardar (2015) [31]	lakota zaspanost napihnjenost bolečina neudobje (mokra planica /hladno/vroče)	LFCC	VQ	91.58%	CA
Bănică et al. (2016) [32]	napihnjenost podrigavanje neudobje lakota bolečina utrujenost	MFCC	GMM i-vectors	58%	CA
Chang & Li (2016) [33]	lakota bolečina zaspanost	FFT Spectrogram	CNN	78.50%	CA
Wahid et al. (2016) [34]	bolečina lakota	MFCC LPCC	MLP RBFN	93.40%	CA

Iz pregleda sorodnih del lahko izpeljemo številne izsledke:

- Čeprav potekajo raziskave na področju avtomatične klasifikacije že več kot 20 let, je področje slabo raziskano. Za ta čas smo našli 26 raziskav, od katerih je več kot polovica bila narejenih v preteklih petih letih.
- Mel frekvenčni kepstralni koeficienti (MFCC) in linearno napovedni kepstralni koeficienti (LPCC) so najpogostejši postopki za pridobitev značilnk. Verjetno so najpogosteje uporabljeni, saj so se izkazali za učinkovite v različnih sistemih avtomatične prepoznave govorcev [35]. To je skladno s pregledom klasifikacij patoloških jokov, ki prav tako poročajo o najpogostejši uporabi MFCC in LPCC [7].
- Skozi leta opazamo trend klasifikacije v vedno več razredov, kar je skladno s kompleksnimi razlogi za jokom, a hkrati težje izvedljivo.
- Viri so izredno redkobesedni v opisu pogojev, pod katerimi so bili joki posneti. Tako ni vedno povsem jasno, kaj je povzročilo bolečino ali kak drug razlog za jok in ali so si bili pogoji med različnimi joki istih kategorij podobni. Slednje je presenetljivo, ker že majhne spremembe v povodu za jok vplivajo na akustično sliko joka, npr. jok zaradi bolečine pika se razlikuje od joka zaradi bolečine želodca [29]. Jok zaradi napihnjenosti v črevesju je drugačen od joka zaradi napihnjenosti v želodcu [32].
- Pregled sorodne literature pokaže, da se znotraj področja klasifikacije dojenčkovih jokov uporablja številne raznolike metode, ni pa povsem razvidno iz uspešnosti klasifikacije, katera je najboljša. Najpogosteje uporabljajo različne vrste nevronske mreže in metodo podpornih vektorjev (SVM) z izborom različnih jedrnih funkcij.
- Raziskave poročajo izredno visoke klasifikacijske natančnosti in z izjemo dveh [21, 24], nobenih drugih mer, ki omogočajo vrednotenje uspešnosti klasifikacijskih modelov. To je problematično, ker je natančnost mera,

ki zamegli resnično uspešnost klasifikacijskih modelov v večrazrednih klasifikacijah. Namreč, velikost natančnosti je sorazmerna z vsoto pravilno uvrščenih tarčnih jokov v tarčni razred in pravilno zavrženega uvrščanja ne tarčnih jokov v tarčni razred. Tako bi model, ki bi natančno uvrščal v razrede, lahko imel visoko natančnost, preprosto z ne uvrščanjem v tarčni razred. Število razredov proporcionalno večja verjetnost uvrščanja ne tarčnih jokov v ne tarčne razrede in seveda natančnosti modela. Hkrati pa model ne dosega kvalitete napovedi tarčnega joka v tarčni razred, ki bi si jo modela želeli. V odsotnosti drugih mer je tako natančnost nezanesljiva mera za primere večrazredne klasifikacije. Slednje onemogoča zanesljivo primerjavo z dosedanja literaturo, kvečjemu hevristično [36].

- Analizirane raziskave zagotavljajo kvaliteto modelov z delitvijo na učno in testno množico ter prečnim preverjanjem. To prepreči preveliko prilaganje modela na učne primere. Za primere, ki jih uvrščajo v testno ali učno množico, uporabijo 1-sekundne posnetke, ki jih s predobdelavo podatkov (npr. odstranitev šuma) pridobijo iz daljših posnetkov in iz katerih pridobijo značilke (npr. MFCC ali LPCC). Z izjemo [31], noben vir ne poroča ločevanja govorcev na učno in testno množico. Sklepamo, da obstaja velika verjetnost, da so različni 1-sekundni posnetki istih govorcev bili uporabljeni tako v učnih kot testnih množicah primerov, iz česar sledi, da je uspešnost napovednih modelov najverjetneje nižja od poročane.
- Lastnosti dojenčkov so redko opisane ali sploh niso opisane, kljub temu da obstajajo številne individualne in skupinske razlike med otroci. Kar 44% povzete literature ne poroča starosti, skorajda nobena pa sploh otrok, katerih posnetki so bili uporabljeni v gradnji klasifikacijskih modelov.

Glede na pregled sorodnih del se zdi zastavljeni cilj magistrske naloge še posebej obetaven, saj je raziskovanje vpliva starosti na uspešnost klasifika-

cijskega modela slepa pega trenutnih sorodnih del. Tak zaključek podpirajo tudi (redke) raziskave iz področja patološke klasifikacije, kjer se je klasifikacijski model za razreda zdrav jok in jok otroka z volčjim žrelom izboljšal za 5-7% ob vključitvi starosti dojenčkov [37, 38].

# Poglavje 3

## Zajem podatkov

Zajem podatkov je bil narejen s ciljem pridobitve zvočnih posnetkov dojenčkovih jokov, ki so akustično bogati s podatki oziroma signalom joka in čim manjšimi nivoji ozadnjega šuma. Posnetki bi morali biti označeni tako s starostjo dojenčka kot z razlogom za jok (razred v katerega klasificiramo) ter potencialno drugimi podatki, ki bi pomagali pri klasifikaciji (npr. spol, opombe, itd.).

### 3.1 Baze podatkov

Obstajajo številne baze podatkov dojenčkovih jokov, kot so na primer *Infant Cry and Causes (ICC) Database* [39], baza *Baby Chillanto*, baza jokov *Soroka* zdravih dojenčkov med zdravniškimi pregledi, ki je posebno zanimiva, saj vključuje starosti otrok [37], baza dojenčkovih jokov iz Urgentne Bolnišnice Sv. Pantelimona [40], baza *OxVoc* [41], ki poleg dojenčkovih jokov vključuje še vokalizacije odraslih in domačih živali, podatkovna baza MPAT [42] s posnetki dojenčkov, ki jočejo zaradi bolečine, ter številne druge baze.

Raziskovalce, ki so zgradili zgoraj omenjene baze, smo zaprosili za dostop do posnetkov izključno za namene magistrskega dela, vendar nismo dobili nobenega odgovora. Pomanjkanje odprto dostopne baze podatkov, ki je sestavljena iz dojenčkovih jokov in drugih podatkov (starosti, spola, razloga

za jok, teže otroka, zdravstvenega stanja, itd.) je problematika s katero se trenutno soočajo raziskovalci tega področja, saj številne raziskave niso narejene dovolj natančno zaradi pomanjkanja podatkov in težavnosti sestavljanja zelene baze [43].

## 3.2 Objavljeni video posnetki

Soočeni s pomanjkanjem podatkovnih baz in dostopa do njih smo se odločili, da bomo preiskali prosto dostopne baze podatkov, kot je na primer Youtube. Youtube ima več kot milijardo uporabnikov in skorajda tretjino celotne svetovne internetne populacije [44], kar kaže, da je dovolj velika baza, kjer bi lahko našli iskane posnetke dojenčkovih jokov. Hkrati Youtube deluje po principu *Fair use*, ki dovoljuje uporabo posnetkov, podatkov in na sploh intelektualne lastnine za akademske in raziskovalne namene brez izrecnega soglasja lastnikov intelektualne lastnine [45].

### 3.2.1 Iskanje podatkov

V iskanju posnetkov joka smo želeli pridobiti zvočno čimbolj čiste in jasne posnetke, brez ozadenjskih šumov ali drugih zvokov. Hkrati smo želeli pridobiti posnetke, kjer so razlogi za jok objektivno podani. Na primer, če je na posnetku dojenček začel jokati med cepljenjem takoj po piku z iglo, smo tak posnetek lahko uvrstili v razred joka zaradi bolečine. Cilj je bil pridobiti čimveč različnih posnetkov, na katerih je dojenčkov jok zvočno čist, razlog za jok jasn in starost dojenčka na posnetku znana iz opisa posnetka.

Za iskanje posnetkov po kanalu smo uporabili iskalne nize opisane v Tabeli 3.1. Če posnetek ni imel jasno označene starosti dojenčka v času snemanja posnetka, je bil dojenček na posnetku starejši od dvanajstih mesecev, posnetek ni podal enega ampak več ali nič razlogov za jok, je bil razlog za jok v opisu očitno drugačen od vsebine posnetka, ali je posnetek bil zvočno in drugače oporečen, ga nismo vključili v našo bazo podatkov.



**Tabela 3.1:** Iskalni nizi za zajem posnetkov iz Youtube-a

Kaj	Iskani niz
dojenček	baby, bebe, bb, infant, babies, baby's, newborn,
jok	cry, crying, tears,
starost	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, month(s), mo, m, week(s), w, day(s), old
jeza	tantrum, angry, fit, frustrated
bolečina	needle, (doctor) vaccination(s), shot(s), teething, pain
strah	fear, affraid, scared, startled, freaked (out), [+Elmo +toy +dad +mean]
zaspanost	tired, sleep(y), fatigue(d), overtired, nap, crib,
mokre	diaper [+poopy +wet +dirty +pee]
planice	

### 3.2.2 Opis posnetkov joka

V bazo posnetkov dojenčkovih jokov (z znanim razlogom za jokom in starostjo dojenčka) smo uvrstili 112 unikatnih posnetkov v skupnem trajanju 1293 sekund. Tabela 3.2 prikazuje, koliko posnetkov je bilo zbranih za določeno kategorijo, kjer smo želeli imeti najmanj 100 sekund posnetkov v vsaki kategoriji. Tabela 3.3 pa prikazuje število posnetkov in skupno trajanje let glede na starost dojenčkov in razlog za jok.

Podroben opis zajetih posnetkov, z unikatnim številom posnetka, starostjo dojenčkov, njihovim spolom, razlogom za jok in trajanjem posnetka se nahaja v prilogi C.

## 3.3 Zajem podatkov z aplikacijo Android

Za zbiranje vzorcev dojenčkovega joka smo razvili Android aplikacijo z imenom Jokolov in jo objavili na Google play trgovini. Starši ali skrbniki otroka so s pomočjo aplikacije Jokolov snemali jok otroka in označili vzrok joka. Navodila znotraj aplikacije so vzpodbujale uporabnika, da označi *vzrok* joka šele po tem, ko je otroka pomiril in je prepričan, kaj je bil vzrok joka. Podatki

**Tabela 3.2:** Posnetki glede na razlog joka

Razred	Število posnetkov	Trajanje [sek]
bolečina	23	416
izločanje	10	250
jeza	13	446
lakota	18	1106
strah	31	168
zaspanost	17	347
SKUPAJ	112	1293

**Tabela 3.3:** Število posnetkov in njihovo trajanje glede na razlog za jok in starost otroka

Kategorija	Mera	Starost [meseci]		
		0-3	3-6	6-12
bolečina	št.	5	13	5
	sek	47	215	154
izločanje	št.	8	2	0
	sek	188	62	0
jeza	št.	2	5	6
	sek	139	136	171
lakota	št.	16	2	0
	sek	1010	96	0
strah	št.	7	8	16
	sek	45	35	88
zaspanost	št.	7	9	1
	sek	103	238	6
Skupno	št.	45	39	28
	sek	532	782	419



**Slika 3.1:** Primer iz aplikacije Jokolov.

o starosti otroka, unikatno številko otroka, oznako in posnetek otroka smo nato s telefona prenesli na REST aplikacijo na OpenStack oblaku. Pogled, kjer uporabnik snema jok, vidimo na sliki 3.1.

Prva različica aplikacije Jokolov je bila na trgovini Google play objavljena 27. maja 2016. Starše, ki bi snemali svojega dojenčka kako joče smo skušali pridobivati osebno preko poznanstev in metode snežne kepe, kjer znanci predlagajo svoje znance. Posnetke prvotnih staršev (tri družine), ki so bili pripravljeni sodelovati smo zaradi napake v aplikaciji izgubili (na strežniku so se shranili posnetki brez zvoka). Po posodobitvi aplikacije samo nekaj dni kasneje nismo uspeli pridobiti staršev, ki bi bili pripravljeni snemati svoje otroke.



## Poglavje 4

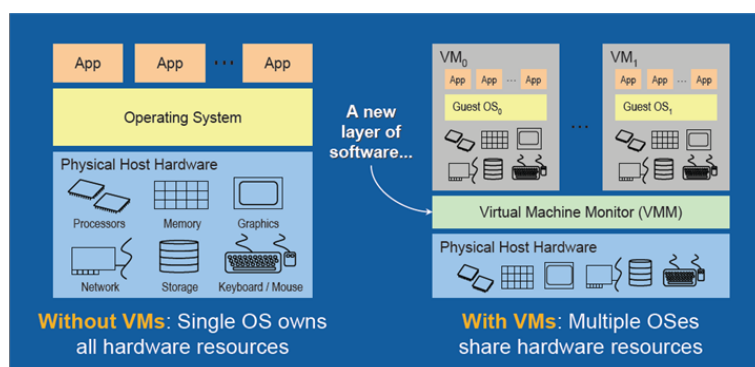
# Hramba in obdelava podatkov v oblaku

### 4.1 Kaj je oblak

Dnevna uporaba osebnega računalnika v povprečju doseže nizko izrabo virov. Običajno dosežemo 10% uporabe procesorske moči, 60% pomnilnika in 20% internetne širine. Če bi upoštevali povprečje uporabe celotnega delovnega dne bi bile številke še nižje [46].

Uporaba strežnikov kaže večinoma podobne vzorce. Zamisel oblaka je v optimizaciji neizkoriščenih virov, ker če bi vire strežnikov združili, bi na tak način plačali samo za vire, ki jih uporabljamo. Nacionalni inštitut standardov in tehnologije (angl. National Institute of Standards and Technology) definira računalništvo v oblaku s to definicijo:

Računalništvo v oblaku je model, ki omogoča vseprisoten, priročen, interneti dostop do skupnih konfigurabilnih računalniških virov (npr. omrežja, strežniki, shranjevanje, aplikacije in storitve) na zahtevo. Viri so hitro omogočeni in dostopni z minimalnim naporom upravljanja ali posredovanja ponudnika storitev. Ta model promovira dostopnost in je sestavljen iz petih ključnih značilnosti, treh modelov za storitve in štirih modelov za namestitve [47].



Slika 4.1: Primerjava stroja z in brez virtualizacije [48].

Preden opišemo ključne značilnosti oblaka je za njegovo razumevanje nujno, da razumemo virtualizacijo, ki omogoča tehnično izvedbo oblaka [46].

## 4.2 Virtualizacija

Virtualizacija je kombinacija programske in strojne opreme, ki omogoča, da ustvarimo navidezni stroj (angl. virtual machine). Z navideznim strojem naredimo abstrakcijo strojne opreme na tak način, kot da bi na mestu vsakega posameznega stroja uporabljali več strojev [48]. Primerajvo stroja z in brez virtualizacije vidimo na sliki 4.1.

### 4.2.1 Hipervisor

Vsak operacijski sistem, ki ga namestimo v navidezni stroj, deluje kot, da so viri (procesor, pomnilnik, disk,...), s katerimi upravlja, pod njegovim nadzorom, čeprav so v resnici v lasti gostiteljskega stroja. Navidezni stroj prav tako nima mehanizma, ki bi mu omogočal, da se zaveda, da je v resnici navidezni stroj in ne stroj nameščen direktno na strojni opremi (angl. bare metal machine).

Funkcionalnost, ki navideznemu stroju omogoča, da deluje kot da bi lahko upravljal s gostiteljskimi viri, ne da bi se tega zavedal (ali brez potrebnih

konfiguracij operacijskega sistema), omogoča nadzornik navideznih strojev (angl. Virtual machine monitor, VMM) ali hipervizor (angl. hypervisor). Hipervizor nadzira gostiteljske vire in navideznim strojem omogoča dostop do virov. Hipervizor omogoča, da navidezne mašine dostopajo do iste strojne opreme in da pri tem ne pride do konfliktov pri dostopu.

Stroj, na katerem hipervizor omogoča delovanje navideznih strojev, imenujemo gostiteljski stroj (angl. Host machine). Vsak navidezni stroj pa imenujemo gostujoči stroj (angl. Guest machine).

### 4.3 Ključne značilnosti

Ključne značilnosti [47]:

- Samopostrežna storitev na zahtevo (angl. On-demand self-service). Uporabnik lahko enostransko in avtomatično ustvari oz. rezervira vire, kot so internetna širina in uporaba strežnika, brez posredovanja ponudnika storitev.
- Širokopasovni mrežni dostop (angl. Broad network access). Viri so dostopni preko omrežja.
- Združevanje virov (angl. Resource pooling). Ponudnikovi viri so združeni na tak način, da lahko do njih dostopajo različni uporabniki. Do iste skupine virov lahko uporabniki dostopajo z različno dodeljenimi fizičnimi in navideznimi viri, ki si jih lahko posamezni uporabniki dinamično dedelijo na zahtevo. Uporabnik nima informacije ali kontrole nad točno lokacijo virov. Uporabnik ima lahko možnost, da na višjem nivoju abstrakcije določi lokacijo (npr. Država, pokrajina, podatkovni center). Primeri virov vključujejo shranjevanje, procesiranje, spomin in internetno širino.
- Hitra elastičnost (angl. Rapid elasticity). Viri so lahko hitro omogočeni in dostopni, v nekaterih primerih se količina virov avtomatično spreminja sorazmerno s povpraševanjem. Uporabniku se pogosto zdi, da je

količina virov, ki so na voljo neomejena in jih lahko uporabi v poljubni količini ob poljubnem času.

- Merjena storitev (angl. Measured service). Oblačni sistemi avtomatično nadzirajo in optimizirajo uporabo virov glede na tip storitve (npr. Shranjevanje, procesiranje, pasovna širina in število aktivnih uporabnikov). Poraba virov je lahko nadzorovana, kontrolirana in predstavljena na način, da zagotavlja transparentnost uporabniku in ponudniku storitve.

## 4.4 Modeli storitve

- Programska oprema kot storitev (angl. Platform as a Service, SaaS).
- Platforma kot storitev (angl. Platform as a Service, PaaS).
- Infrastruktura kot storitev (angl. Infrastructure as a Service, IaaS).

## 4.5 Modeli namestitve

**Privatni oblak** oblačno strukturo uporablja izključno ena organizacija v eni ali večih poslovnih enotah. Upravlja jo lahko organizacija sama ali druga organizacija.

**Skupnostni oblak** oblačno strukturo uporablja omejeno število uporabnikov. Običajno so to organizacije s skupnimi interese. Upravljajo jo lahko organizacije same ali druge organizacije.

**Javni oblak** uporaba oblačne strukture je omogočena javnosti.

**Hibriden oblak** kombinacija dveh ali večih ostalih modelov namestitve.



## 4.6 OpenStack

OpenStack je odprtokodna oblachna platforma za javne in privatne oblake, ki lahko nadzira veliko množico virov. Vire lahko upravljamo preko grafičnega vmesnika ali preko OpenStack APIja. OpenStack platforma je sestavljena iz šestih jedrnih komponent:

**NOVA** Nova Compute je komponenta, ki skrbi za računsko infrastrukturo. Odgovorna je za zagon, časovno planiranje in uničenje strojev.

**NEUTRON** Neutron Networking je komponenta, ki skrbi za omrežno infrastrukturo. Neutron komponenta omogoča povezljivost za ostale komponente OpenStacka (npr. Nova). Neutron komponenta nam omogoča uporabo APIja za definiranje omrežji.

**SWIFT** Swift Object Storage je komponenta, ki omogoča in je odgovorna za nadzor shranjevanja objektov. S pomočjo Swift komponente lahko shranjujemo in pridobimo objekte preko REST HTTP APIja. Zaradi možnosti replikacije podatkov in skalabilnosti je shranjevanje objektov s Swift komponento zanesljivo.

**CINDER** Cinder Block Storage omogoča shranjevanje podatkov za stroje.

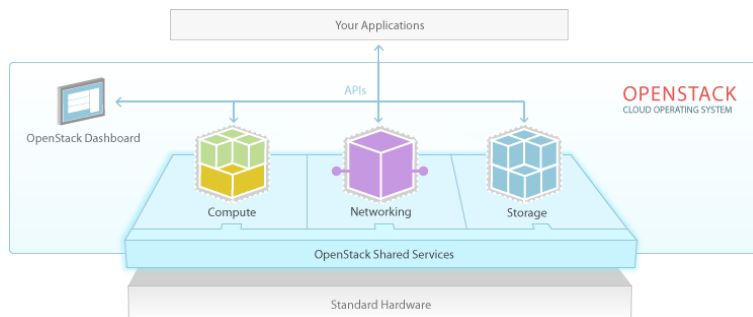
**KEYSTONE** Keystone Identity je komponenta, ki omogoča avtorizacijo in avtentikacijo za ostale komponente OpenStacka.

**GLANCE** Glance Image Service je komponenta, ki nam omogoča, da pridobimo in shranjujemo slike virtualnih strojev.

Diagram OpenStack platforme vidimo na sliki 4.2.

## 4.7 Django REST API

Posnetek o joku zajemamo z Android aplikacijo Jokolov in ga po zajetju pošljemo na Django strežnik. Na strežniku posnetek obdelamo in klasificiramo v nek razred glede na predlog napovednega modela. Klasifikacijska

**Slika 4.2:** Diagram OpenStack platforme

Vir: <https://www.openstack.org/>

oznaka se nato prenese nazaj aplikaciji Android. Za pošiljanje in branje podatkov s strežnika smo uporabili standardizirano metodo. Uporaba standardizirane metode omogoča večjo fleksibilnost in varnost. Če se bomo kdarkoli v prihodnje odločili, da razvijemo tudi iOS aplikacijo, smo z uporabo standardiziranega vmesnika za prenos podatkov prihranili veliko časa.

Za prenos podatkov smo razvili vmesnik uporabniškega programa REST (ang. REpresentational State Transfer) API (angl. Application Programming Interface). Način delovanja REST APIja je zelo podoben načinu delovanja spletnega strežnika. Klient (v našem primeru aplikacija Jokolov) pošlje (preko http ali https protokola) zahtevo na strežnik (v našem primeru Django strežnik). Strežnik nato odgovori s številko statusa in pri določenih zahtevah tudi s podatki.

Dobra praksa narekuje, da odgovore od API strežnika dobimo v enem izmed standardnih formatov (XML, JSON, ...). V naši implementaciji Django REST API strežnika podatke dobivamo v JSON (JavaScript Object Notation) formatu. JSON podatke dobivamo v obliki ključ-podatki, kar pomeni, da do vsakega podatka (ali seznama podatkov) dospemo z neko ključno besedo.

REST API strežnik nam omogoča, da pri zahtevi na strežnik uporabimo parametre, kot so recimo ID (recimo za dostop do specifičnega posnetka) ali older\_by (recimo za dostop do vseh posnetkov, ki so starejši od nekega

datuma).

Do podatkov dostopamo s http (ali https) metodami. Django REST API omogoča GET, POST, PUT, PATCH in DELETE metode:

GET S pomočjo GET metode pridobimo podatke s strežnika. V primeru, da pri zahtevi ne uporabimo parametra, nam strežnik vrne seznam vseh objektov (recimo posnetkov). Če pri zahtevi uporabimo parameter pk nam strežnik vrne točno določen zapis (recimo posnetek).

POST S pomočjo POST metode podatke pošljamo na strežnik. Podatke ne pošljamo s pomočjo parametrov, ampak jih zapišemo v telo (angl. body) zahtevka. Podatki, ki jih pošljamo s v JSON formatu. Podatke lahko pošljamo v seznamu (več hkrati) ali vsakega posebej.

PUT S pomočjo PUT metode zamenjamo vse podatke nekega objekta z na novo poslanimi podatki. V parametru specificiramo kateri objekt želimo zamenjati (s parametrom pk).

PATCH S pomočjo PATCH metode popravimo oz. posodobimo objekt. Pri tej metodi nam ni potrebno skrbeti, če so kakšni podatki na objektu že nastavljeni, mi pa želimo posodobiti samo nekatere. Pri metodi PUT pa moramo biti pozorni saj se vsi podatki, ki jih ne pošljemo v zahtevi izbrišejo. V parametru specificiramo kateri objekt želimo posodobiti (s parametrom pk).

DELETE S pomočjo metode DELETE izbrišemo objekt. V parametru specificiramo kateri objekt želimo izbrisati (s parametrom pk).

### 4.7.1 REST API VARNOST

Django REST API nam omogoča, da poskrbimo za osnovne principe informacijske varnosti kot so zaupnost, neokrnjenost in razpoložljivost.

## **Zaupnost**

Princip zaupnost govori, da moramo za podatke, ki so zaupne narave (ime, priimek, geslo, datum rojstva, vsebina posnetka, ...) poskrbeti, da jih lahko vidijo le pooblašcene osebe. Zaradi tega principa Django REST API poskrbi da:

- Vsak objekt poslan na strežnik ima zabeleženega avtorja (kdo je naredil zahtevo).
- Samo avtenticirani uporabniki lahko ustvarijo nov objekt.
- Samo avtor objekta lahko spremeni ali izbriše objekt.
- Ne avtenticirani zahtevki imajo omejen dostop do podatkov. Podatke do katerih lahko dostopajo, smejo samo pogledati ne pa tudi spreminjati [49].

## **Avtentikacija**

Avtentikacija je način kako zahtevo povežemo z njenim avtorjem. Django REST API nam omogoča naslednje avtentikacije:

- `BasicAuthentication` je implementacija osnovne HTTP avtentikacije (angl. HTTP Basic Authentication), ki je specificirana v RFC (request for Comments) standardu [50]. Pri osnovni avtentikaciji se avtenticiramo s pomočjo uporabniškega imena in gesla. Zaradi prenosa občutljivih podatkov je v produkciji pomembno, da poskrbimo, da so podatki ob prenosu kriptirani (uporaba https protokola). Pri ne avtenticirani zahtevi nam strežnik vrne odgovor s kodo 401.
- `TokenAuthentication` je implementacija enostavne token avtentikacije (angl. Simple token-based HTTP Authentication). Pri token avtentikaciji se avtenticiramo z dolgim naključnim nizom, ki je unikatni za vsakega uporabnika. Pri enostavni token avtentikaciji se nizi ne spreminjajo in za posameznega uporabnika ostaja isti za cel čas njegove

uporabe. Zaradi tega, ker se niz ne spreminja je tudi pri tej avtentikaciji nujno, da v produkciji poskrbimo, da so podatki ob prenosu kriptirani (uporaba https protokola).

- SessionAuthentication za avtentikacijo uporablja Djangove seje. SessionAuthentication je primerna za AJAX kliente, ki komunicirajo s spletno stranjo znotraj iste infrastrukture.
- Implementacija svoje avtentikacije ali vključitev drugih knjižnic za avtentikacijo (npr. OAuth avtentikacija).

## 4.8 Django

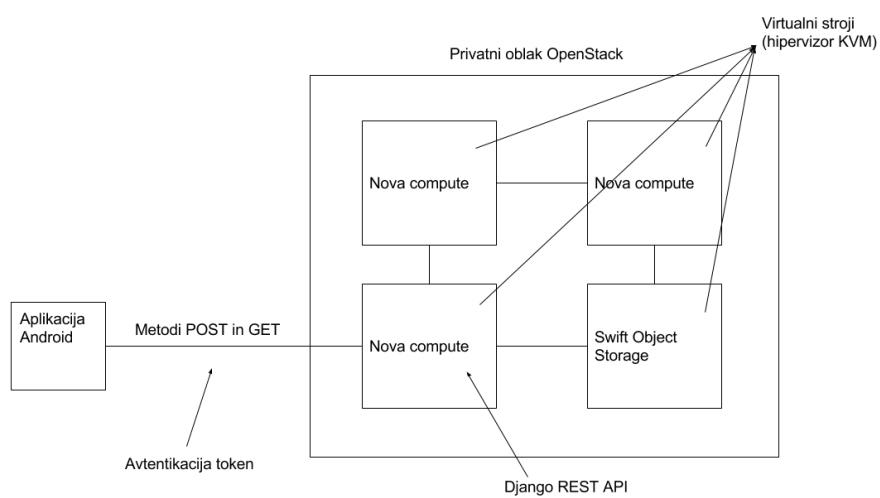
Django je spletna aplikacijska platforma (angl. Web Application Framework). Javno je bil objavljen leta 2005, vendar je postal popularen po objavi verzije 1.0 leta 2008. Django omogoča veliko vgrajenih rešitev v sam framework. Primeri takih rešitev so avtentikacija, administracija vsebine, RSS,...

Prav tako smo z uporabo Djanga primorani upoštevati osnovne varnostne smernice in se na tak način izognemo pogostim varnostnim napakam. Način kako Django deluje preprečuje napade na zbirke podatkov ob pomoči vrivanja SQL (angl. SQL injection), XSS (cross-site scripting), ponarejanje zahtev iz drugih strežnikov in ugrabitev seje. Prav tako skrbi za varno upravljanje z uporabniškimi računi in njihovimi gesli.

API, ki skrbi za shranjevanje posnetkov in pridobivanje podatkov smo implementirali v Django platformi in API namestili na OpenStack oblak.

## 4.9 Uporabljena infrastruktura

Na sliki 4.3 vidimo uporabljeno infrastrukturo za obdelavo in zajem posnetkov z aplikacijo Android. Posnetki se prenesejo z aplikacije Android preko metode POST na Django REST API na privatnem oblaku OpenStack.



**Slika 4.3:** Uporabljena infrastruktura za obdelavo in zajem posnetkov z aplikacijo Android

# Poglavje 5

## Napovedni model razloga za jok

Področje strojnega učenja predstavlja uporabo statističnih in računalniških tehnik, ki so usmerjene k cilju pridobivanja koristnih informacij iz še ne (zadostno) obdelanih podatkov. V tem delu je strojno učenje bilo uporabljeno, da na podlagi množice posnetkov predobdelanih dojenčkovih jokov (z ustreznimi oznakami vzrokov za jok, dojenčkovo starostjo in spolom) razvijemo model, ki je sposoben napovedati razlog dojenčkovega joka iz novega - še ne prej obdelanega - zvočnega posnetka dojenčkovega joka.

### 5.1 Zgradba modela joka

Model je bil zgrajen v več korakih:

1. Obdelava posnetkov. Pretvorimo surove posnetke joka v primere za analizo.
2. Pridobivanje značilnk. Iz primerov pridobimo značilke za sestavo učnih in testnih podatkov za napovedni model.
3. Učenje modela za prepoznavanje joka. Iz danega zvočnega signala naučimo model izbrati zgolj signal joka in ne okoljskega šuma za objekt podatkovnega rudarjenja. Učenje skozi binarno klasifikacijo v razreda jok in ne-jok.

4. Klasifikacija joka v šest razredov joka. Klasifikacija na podlagi k-najbližjih sosedov, metode podpornih vektorjev, naključnega gozda, logistične regresije in rekurenčne nevronske mreže. Modele med seboj primerjalno vrednotimo po natančnosti in izberemo najboljšega.
5. Za najboljši klasifikacijski model preverimo, ali je starost dojenčkov izboljšala napovedno uspešnost modela.

## 5.2 Pridobivanje značilk

V našem primeru smo morali ločiti signal dojenčkovega joka od ostalih okoljskih šumov v zvočnem posnetku. Predobdelava zvočnega posnetka je potrebna za nadaljno analizo klasifikacije atributov joka v razrede razlogov za jok, saj bi v nasprotnem primeru hkrati z atributi joka uvrščali v razrede tudi šum v posnetkih. Predpostavka tega pristopa je, da so zgolj zvočni atributi joka in ne šum zadostno in konsistentno ponovljivi, da omogočajo napovedovanje pripadnosti razredu. Napovedni model smo naučili uvrščati zvočne posnetke v dva razreda: take, ki vsebujejo dojenčkov jok, in take, ki joka ne vsebujejo. V prilogi A razložimo kakšne lastnosti ima zvočni signal pri joku in kako se razlikuje glede na posameznike in skupine.

### 5.2.1 MFCC

Mel frekvenčni kepstralni koeficienti (angl. Mel-Frequency Cepstral Coefficients, MFCC) so v področju avtomatične klasifikacije zvočnih signalov med najbolj pogosto uporabljenimi značilkami, poleg linearnih napovednih koeficientov (angl. Linear Prediction Coefficients LPCs) in linearnih napovednih kepstralnih koeficientov (angl. Linear Prediction Cepstral Coefficients, LPCCs). Različni avtorji so jih že uporabili v sistemih avtomatične klasifikacije zvoka kot na primer za samodejno razbiranje števil iz posnetka telefonskega pogovora [51], avtomatično klasifikacijo glasbe v žanre [52], predvsem pa klasifikacijo dojenčkovega joka kot patološko ali normalno [53, 54]. Slednje



kaže, da bi bili primerna značilka za klasifikacijo dojenčkovega joka glede na njegov vzrok.

MFCC izračunamo po sledečih korakih [35, str. 314-423]:

1. Zvočni posnetek vzorčimo na 25 ms trajajoče okvirje, ki jih postavimo na vsakih 10ms originalnega posnetka (vzorci se delno prekrivajo). Izbrana dolžina okna prinaša v povprečju zanesljivo spektralno oceno vsebovanih frekvenc. Krajše okno ne bi bilo vzorčno reprezentativno, medtem ko bi daljše okno vsebovalo preveč variacij v vsebovanih frekvencah. Na vsakem oknu bomo generirali mel frekvenčne spektralne koeficiente.
2. Da bi pridobili informacijo o frekvenčni sestavi vsakega vzorca, za vsako okno izračunamo diskretno Fourierjevo transformacijo (DFT)  $X_a[k]$ :

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} \quad (5.1)$$

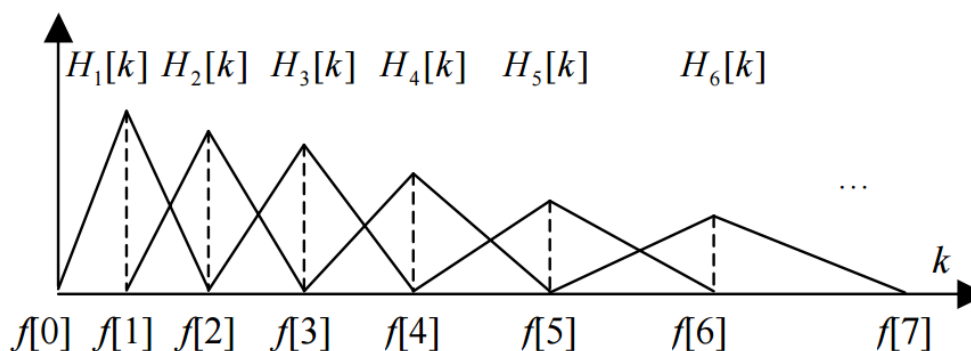
kjer je  $X_a[k]$  DFT,  $k$  frekvenca, ki jo analiziramo,  $n$  vzorec oziroma okno na katerem izvajamo DFT,  $N$  število vseh časovnih oken,  $x[n]$  pa vrednost signala za časovno okno  $n$ .

Nato izračunamo oceno močnostnega spektra s pomočjo periodograma za vsak DFT, ki identificira prisotnost določene frekvence v diskretni Fourierjevi transformaciji:

$$P_a[k] = \frac{|X_a(k)|^2}{N} \quad (5.2)$$

3. Na močnostni spekter apliciramo Mel zaporedje pasovnih filtrov in seštejemo energije znotraj vsakega pasu.

Mel lestvica, na kateri gradimo zaporedje pasovnih filtrov, se razlikuje od linearnih frekvenčnih lestvic uporabljenih v LPC ali LPCC v razmaku med soslednjimi frekvencami, kjer je na Mel lestvici razmak pseudologaritemski ne pa linearni. Slednje ustreza človeški zaznavi frekvenc,



**Slika 5.1:** Trikotni filtri mel pasovnih širin [35, str. 314]

saj nižje frekvence medsebojno ločujemo bolje od višjih. Ta lastnost človeške frekvenčne zaznave izhaja iz biološke zgradbe našega slušnega organa polža, ki je v spiralni votlini posejan z zvočnimi čutnicami v razmakih podobnimi Mel lestvici. Ko zvočni signal pride v uho, zaniha čutnice v tistem pasu polža, ki ustrezajo frekvenci zvočnega signala. Ker so čutnice gosteje posajene v delu pasu za nižje frekvence, te zaznavamo boljše.

Mel zaporedje pasovnih filtrov uporabimo na DFT kot vhodni signal in ustvarimo zaporedje pasovnih filtrov z  $M$  filtri, kjer je poljubni  $m$ -ti filter trikotni filter iz množice  $1, 2, 3, \dots, M$ . Poljubni trikotni filter  $H_m[k]$  določimo kot:

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{2(k-f[m-1])}{(f[m+1]-f[m-1])(f[m]-f[m-1])} & f[m-1] \leq k \leq f[m] \\ \frac{2(f[m+1]-k)}{(f[m+1]-f[m-1])(f[m+1]-f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases} \quad (5.3)$$

Pridobljeni trikotni filtri izračunajo povprečni močnostni spekter znotraj filtra za vsako centralno frekvenco naraščajočih pasovnih širin, kot ponazorijo [35, 315] na sliki 5.1.

4. V naslednjem koraku določimo logaritem za povprečni močnostni spekter vsakega pasovnega filtra:

$$S[m] = \ln \left[ \sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k] \right], \quad 0 \leq m < M \quad (5.4)$$

5. Mel frekvenčni kepstrum izračunamo z diskretno kosinusno transformacijo logaritma močnostnega spektra:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos \left( \frac{\pi n (m + \frac{1}{2})}{M} \right), \quad 0 \leq m < M \quad (5.5)$$

Močnostni spekter različnih filtrov močno korelirajo med seboj, ker so zaporedja pasovnih filtrov prekrivajoča. Diskretna kosinusna transformacija dekorelira medsebojne korelacije.

6. Z diskretno kosinusno transformacijo logaritmov močnostnega spektra pridobimo od 24 do 40 Mel frekvenčnih kepstralnih koeficientov (odvisno od implementacije,  $M$  število filtrov varira od 24 do 40). Za namene avtomatične klasifikacije zvočnih signalov ponavadi ohranimo zgolj prvih 13 koeficientov, saj višji koeficienti ponavadi nimajo velikega informacijskega prispevka v modelih avtomatične klasifikacije.

Pred izračunom vektorja značilnk z MFCC, moramo biti pozorni, saj MFCC niso odporni na učinke šuma [55]. Tygi in Wellekens [55] zato predlagata normalizacijo zvočnega signala pred izborom značilke MFCC ali pa uporabo logaritma večjega razreda v četrtem koraku izpeljave MFCC.

MFCC predstavljajo analizo spektralne moči določenega časovnega okvirja in so posledično statična reprezentacija zvočnega signala. A za zvočni signal je velikokrat pomembna ravno njegova dinamičnost, tj. kako se odvija in spreminja skozi čas, kako frekvence naraščajo ali padajo po časovni osi.

V ta namen velikokrat izračunamo delta MFCC in delta-delta MFCC, ki sta množica koeficientov zgrajenih na MFCC in odražata spreminjanje

MFCC vrednosti znotraj izbranega okna. Delta MFCC  $d_t$  bi izračunali za  $n$ -ti Mel frekvenčni kepstralni koeficient po sledeči formuli [35]:

$$d_t = \frac{\sum_{n=1}^N n (c_{t+n} - c_{t-n})}{2 \sum_{n=1}^N n^2} \quad (5.6)$$

Ponavadi za kakovostne delta MFCC zadostuje že  $n=1$ , kar formulo poenostavi v:

$$d_{t(\text{poenostavljeno})} = \frac{c_{t+1} - c_{t-1}}{2} \quad (5.7)$$

Za vsak Mel frekvenčno kepstralni koeficient pridobimo en delta Mel frekvenčni kepstralni koeficient. Delta-MFCC koeficiente pripnemo v vektor značilnk po MFCC koeficientih. Delta-delta MFCC se izračuna na soroden način kot delta-MFCC, le da vzame za argument v enačbo delta Mel frekvenčni kepstalni koeficient, ne pa Mel frekvenčni kepstralni koeficient.

### 5.3 Preveliko prilagajanje modela

Vsak napovedni model se uči klasificirati na podlagi primerov v učni množici. Kar se je naučil, uporabi na testni množici za preverjanje pravilnosti svoje klasifikacije. Če se isti primeri pojavijo v učni in testni množici, bo model imel visoko natančnost klasifikacije, ampak se bo preveč prilagodil učnim primerom in ne bo dobro klasificiral nove, še nikoli videne, primere. Da bi preprečili preveliko prilagajanje modela na učne primere smo poskrbeli, da posnetki istega otroka niso v učni in testni množici.

### 5.4 Opis metod (večrazrednih) klasifikacij

Vse metode klasifikacije, ki jih opisujemo v tem delu, se učijo na podlagi učnih primerov (podmnožica vseh posnetkov, kjer je vsak primer obdelan del posnetka, označen z razredom, kateremu pripada) uvrščati testne primere v ciljne razrede. Taki obliki učenja pravimo nadzorovano učenje, saj klasifikatorjem povemo, kateri so tarčni razredi. Naši primeri so obdelani

posnetki joka, ki jih klasificiramo v enega izmed šest razredov. Razredi so razlogi za jok. Raredi so: strah, lakota, zaspanost, jeza, mokra plenica in bolečina. V tem razdelku opišemo delovanje vsakega klasifikatorja.

### 5.4.1 Metoda k-najbližjih sosedov

Metoda k-najbližjih sosedov klasificira testne primere v isti razred kot (utežena) večina njihovih k sosedov glede na oddaljenost testnih od učnih primerov v vektorskem prostoru značilk.

Učni primer lahko napišemo v obliki vektorja  $\mathbf{v}$ , ki ima za komponente attribute  $x_i$  in pripadajoči razred  $y$ . Formalno lahko primer z  $n$  atributi ponazorimo kot transponirani vrstični vektor  $\mathbf{v} = [x_1, x_2, x_3, \dots, x_n, y]^T$ . Množica vseh učnih primerov tvori vektorski prostor atributov z  $n$  dimenzijami. Algoritem k najbližjih sosedov uvršča testne primere glede na oddaljenost njihovih atributov od atributov učnih primerov v vektorskem prostoru. Metoda pripiše testnemu vektorju tisti razred, ki ga ima večina njegovih k sosedov v določeni oddaljenosti od testnega vektorja.

Implementacija k- najbližjih sosedov je torej odvisna od trojega, (1) izbranega števila k sosedov, ki so relevantni v glasovanju za pripadnost razredu, (2) določitev mere oddaljenosti s katero pada vpliv sosedstva in (3) vplivom na način glasovanja (preprosto večinski ali utežen).

V implementaciji k-najbližjih sosedov je oddaljenost v prostoru mera, ki ponazarja podobnost ali različnost med vektorji. Bolj sta si vektorja podobna, bolj bodo njuni atributi blizu v prostoru, saj bodo zasedali podobne vrednosti. Najpogosteje se uporabljata dve meri oddaljenosti  $d$  v vektorskem prostoru:

1. Evklidska razdalja v kartezičnem prostoru med dvema točkama  $a$  in  $b$ :

$$d(a, b) = \sqrt{\sum_{i=1}^k (a_i - b_i)^2} \quad (5.8)$$

2. Manhattanska razdalja v kartezičnem prostoru med projekcijama dolžin

$n$ -dimenzionalnih vektorjev  $p$  in  $q$  na koordinatni osi:

$$d(p, q) = \sum_{i=1}^n |p_i - q_i| \quad (5.9)$$

V višjedimenzionalnih prostorih se mere razdalje obnašajo drugače od našega intuitivnega razumevanja, ki je osnovan na tridimenzionalnem prostoru. Evklidska razdalja postane kmalu neuporabna, ker v višjedimenzionalnih prostorih slabo ločuje med različnimi vektorji. Zato je bolj priporočljivo uporabiti Manhattansko razdaljo [56].

Večja frekvenca določenega razreda primerov v učni množici lahko močno vpliva na napovedno uspešnost algoritma. Pod predpostavko, da so atributi delno prekrivajoči med primeri različnih razredov in posledično v istem vektorskem podprostoru značilik najdemo različne razrede, bodo pogostejši primeri z večjo verjetnostjo bili znotraj  $k$  sosedov, ki prispevajo k glasovanju o klasifikaciji še ne uvrščenega primera. Tako glasovanje ne bo osnovano na strukturi atributov določenega razreda, ampak na pogostosti atributov pripadajočim temu razredu. Da bi preprečili nagibanje glasovanja v smeri frekvenčnosti primerov, moramo utežiti primere v učni množici glede na razredno frekvenčnost.

### 5.4.2 Metoda podpornih vektorjev

Metoda podpornih vektorjev (angl. *support vector machine*, SVM) je ne-probabilističen binarni linearni klasifikator, ki v osnovni različici na podlagi označenih učnih primerov klasificira nove primere v enega od dveh razredov.

Učne primere, ki jih želimo klasificirati, zapišemo kot vektorje ( $\mathbf{v}_i$ , kjer so atributi  $n_j$  primerov podani kot elementi vektorja  $\mathbf{v}_i = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \dots, \mathbf{n}_j]$ ). Skupaj vsi podatki tvorijo  $j$ -dimenzionalni vektorski prostor, v katerem je vsak vektor označen s pripadajočim razredom.

Metoda podpornih vektorjev išče hiperravnino med razredoma primerov, ki ima največji rob (angl. *margin*) med najbližjimi primeri obeh razredov [57].

Naloga algoritma je iskanje najkrajšega normalnega vektorja  $w$  in konstante  $b$ , da velja spodnji izraz:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$$

Za primere večrazredne klasifikacije (kot je naš), smo izbrali (LIBVM) implementacijo “ena-proti-ena”. Če je  $k$  število razredov, v katere klasificiramo, metoda “ena-proti-ena” konstruira  $k(k-1)/2$  binarnih klasifikatorjev, ki uporabijo metodo podpornih vektorjev za iskanje hiperravnine z najdaljšo širino robov med dvema razredoma primerov. Vsak klasifikator se upošteva kot en glas za vsak uvrščen podatek, končni podporni vektorji se izberejo na podlagi večine glasov. Čeprav obstajajo številne metode konstruiranja podpornih vektorjev v problemu večrazredne klasifikacije, je “ena-proti-ena” primerjalno uspešna metoda [58].

V primeru večdimenzionalnih ali linearno neločljivih vektorjev atributov uporabimo jedrno funkcijo, ki preslika vhodne podatke v prostor značilk. Poznamo štiri jedrne funkcije: linearno, sigmoidno, RBF in polinomsko.

Matematično lahko zvedemo iskanje hiperravnine z najdaljšim robom na minimiziranje funkcije napake na dva načina, s C-SVM ali *nu*-SVM.

### 5.4.3 Naključni gozd

Naključni gozd je ansambelska klasifikacijska metoda, ki z izgradnjo številnih klasifikacijskih dreves nad množico učnih primerov in korekturo njihovega prevelikega prilagajanja skozi glasovanje dreves, konvergira na končno klasifikacijo testnih primerov v razrede [59].

Klasifikacijska drevesa izračunamo nad učno množico primerov, ki jih predstavimo kot vhodni podatek v obliki seznama primera  $(x, Y) = (x_1, x_2, x_3, \dots, x_n, Y)$ , kjer je  $x_i$  atribut primera in  $Y$  razred, v katerega uvrščamo primer. Obstajajo številni algoritmi nad terkami, ki so uporabljeni v izgradnji klasifikacijskih dreves kot na primer ID3, C4.5, CART, MARS, CHAID itd. Na primer ID3 in kasnejši C4.5 gradita drevesa

na podlagi kriterija informacijske entropije, kjer algoritem določi tisti atribut, ki najboljše razdeli množico primerov na največ homogenih (glede na vrednost izbranega atributa) podmnožic. Začne pri korenu drevesa z vsemi učnimi primeri in razdeli učno množico na podmnožice na vozlišču. Iterativno ponovi delitev množice v vozliču na podmnožice glede na informacijsko entropijo za vsako ustvarjeno podmnožico, dokler delitev po atributih ni več smiselna oziroma mogoča, ker doseže končno informacijsko entropijo [60].

Metoda uvrščanja v razrede s klasifikacijskimi drevesi je problematična, ker ni robustna na spremembe v učnih primerih in pogosto vodi do prevelikega prilagajanja modela na učne podatke [61, str. 315]. Zato jo nadgradimo z metodo naključnih gozdov, ki kot ansambelska metoda zgradi številna klasifikacijska drevesa. Iz množice vseh primerov  $P$  z atributi  $a$  izberemo podmnožico atributov  $a'$ , ki je veliko manjša od množice  $a$  (ponavadi ranga kvadratnega korena  $a$  [62]). Na tej podmnožici atributov zgradimo drevo. Za vsako klasifikacijsko drevo izberemo novo podmnožico atributov izmed vseh atributov. Metoda naključni gozd preide iz dreves na gozd tako da za vozlišče (oziroma atribut po katerem delimo primere v podmnožice) izbere tisto vozlišče, katero je izbrala večina klasifikacijskih dreves.

#### 5.4.4 Logistična regresija

Logistična regresija je klasifikator, ki uvršča realne vektorje v kategorične razrede. Na podlagi kumulativne logistične distribucije (sigmoid) izračuna verjetnost, da nek vhodni atributni vektor pripada danemu kategoričnemu razredu. V primeru večrazredne klasifikacije, logistična regresija pripiše vsakemu vhodnemu vektorju atributov v učni množici določeno verjetnost (imenujemo regresijski koeficient), da bo ta pripadal danemu razredu učne množice. Da bi preprečili preveliko prilagajanje modela učnim primerom uporabimo regularizacijo. Regularizacija je zmanjšanje verjetnosti napake za vsak vhodni vektor in tarčni razred. V praksi uporabljamo dve regularizacijski funkciji, Lasso (ali regularizacija L1) in Ridge (ali regularizacija L2). Razlikujeta se predvsem v tem, da Lasso uvaja popravke prevelikega prilagajanja na zgolj



določenih učnih primerih, medtem ko jih Ridge razporedi čez vse primere.

### 5.4.5 Rekurenčne nevronske mreže

S pomočjo rekurenčnih nevronskih mrež odkrivamo vzorce v podatkih, kot so besedilo, genomi, ročna pisava, govorjene besede ali druge množice podatkov, senzorjev, borz, javnih podatkov,...

Rekurenčne nevronske mreže za razliko od ostalih vnaprejšnjih (angl. feedforward) mrež, za vhod v mrežo ne uporabi samo trenutnih podatkov, ampak tudi podatke, ki so bili na vhodu v prejšnjem koraku. Odločitev rekurentne nevronske mreže v času  $t - 1$  vpliva na odločitev rekurenčne nevronske mreže v času  $t$  [63].

LSTM (long short-term memory) mreže so način kako lahko shranjujemo podatke prejšnjega stanja. LSTM enota si shrani prejšnje stanje in ga v naslednji iteraciji uporabi pri odločitvi [64].

### 5.4.6 Primerjalno vrednotenje uspešnosti napovedovanja razloga za jok

Kontingenčna tabela ali tabela napak nam omogoča, da vrednotimo različne klasifikatorje z enakimi kriteriji [65]. Kontingenčna tabela 5.1 razdeli načine, kako lahko obravnavamo podatke glede na napovedni model (njihovo napovedno vrednost) v primerjavi z njihovimi dejanskimi vrednostmi:

Za določanje in primerjavo uspešnosti klasifikacijskih algoritmov lahko uporabimo številne mere:

- Točnost (angl. accuracy, CA):  $(TP + TN)/(P + N)$
- Natančnost ali delež pravilnih pozitivnih (angl. positive predictive value ali precision, PPV):  $TP / (TP + FP)$
- Priklic, znan tudi kot občutljivost (angl. Sensitivity, true positive rate recall)  $TPR = TP/P = TP / (TP + FN)$

**Tabela 5.1:** Kontingenčna tabela

		Dejanska vrednost		
		+	-	
Napovedana vrednost	+	Zadetek (angl. True Positive, TP)	Lažni alarm (angl. False Positive, FP)	P
	-	Pogrešek (angl. False Negative, FN)	Pravilna zavrnitev (angl. True Negative, TN)	N
		P	N	

- Površina pod krivuljo ROC (area under the receiving operator characteristic curve, AUC)

Točnost (accuracy) je premosorazmerna z velikostjo zadetkov in pravih zavrnitev. V primeru večrazrednih klasifikacij, bi model, ki naključno ali napačno uvršča joke v razrede, lahko dosegel visoko mero pravih zavrnitev. Verjetnost, da bo model pravilno zavrnil umestitev ne tarčnega joka v ne tarčni razred, je sorazmerna s številom razredov: več je razredov, večja je verjetnost, da bo model klasificiral ne tarčni jok v katerega izmed njih. Posledično je točnost nezanesljiva mera kvalitete klasifikacijskega modela, če poleg nimamo še drugih mer, ki lahko razjasnijo ali je velikost točnosti pogojena z zadetki ali pravih zavrnitvami na konkretnih primerih. Ravno zaradi njene neodpornosti na spremembe izven razredov za katere želimo vedeti uspešnost klasifikacijskih modelov, pregledi literature odvrtačajo bralce od uporabe natančnosti brez drugih mer [36].

Mera AUC je še posebej zanimiva in uporabna, ker je odporna na pristranskosti neenakomerno številčnih razredov in neenakomerno porazdeljenih napak. Krivuljo ROC zarišemo na grafu, ki ima za osi PTR (angl. positive true rate, glej zgoraj, delež pravilno napovedanih primerov) in FPR (angl. false positive rate, izračunan kot  $FP/N$ , je delež lažnih alarmov) za vsak klasificiran primer. Diagonala grafičnega prikaza je enakovredna naključni klasifikaciji ( $p = 0.5$ ) in vse krivulje levo nad diagonalo prikazujejo klasifi-

kacijo, ki je bolj uspešna od naključne. Napovedno uspešnost izračunamo iz ROC krivulje kot površino pod krivuljo (AUC), kjer je vrednost AUC 0.5 enaka naključni klasifikaciji, 0.75 in več je uspešna napoved, napovedna uspešnost nad 0.9 pa spada med najvišje [66].

## 5.5 Orange

Klasifikacijske modele smo gradili z odprtokodnim orodjem za podatkovno rudarjenje in vizualizacijo Orange. Osnovan je na C++ in Python ovojnica s številnimi integriranimi odprtokodnimi knjižnicami kot so LIBSVM za metodo podpornih vektorjev, LIBLINEAR, Earth, QHull in delno BLAS ter Pythonove knjižnice numpy, networkx in matplotlib [67].

Pregled odprtokodnih orodij za podatkovno rudarjenje in vizualizacijo pokaže, da so si odprtokodna orodja zelo podobna in ni izrazitega zmagovalca za številne zahteve, ki jih mora tako orodje izpolniti [68]. Poglavitni razlog za našo izbiro orodja Orange v primerjavi z drugimi leži predvsem v hitrosti podatkovne obdelave, preprosti vizualizaciji kompleksnih procesov analize in gradnje modelov ter možnosti skriptiranja v primeru, da bi radi kak model spremenili.



# Poglavje 6

## Metodologija in rezultati

V tabeli 3.2 vidimo, da smo pri našem delu uporabili 112 posnetkov joka v skupni dolžini 1293 sekund. Podroben opis posameznih posnetkov vidimo v prilogi C. Iz tabele 3.2 je tudi razvidna porazdelitev razredov joka. Za binarno klasifikacijo smo razrede joka združili v en razred joka in uporabili ločene posnetke za šum. V tabeli 6.1 vidimo, da smo za šum uporabili 14 posnetkov v skupni dolžini 3612 sekund.

### 6.1 Obdelava posnetkov

Zbranim posnetkom smo odstranili dele, za katere smo lahko jasno označili, da niso del joka (televizija, promet, tišina na začetnem delu posnetka, tišina na končnem delu posnetka, ...). Posnetke smo nato razdelili na posamezne posnetke dolžine ene sekunde, kot je značilno za to področje. Posamezen posnetek z dolžino ene sekunde smo uporabili kot posamezen primer joka.

Zbrani posnetki so bili v formatu flv (Flash Video), ki smo jih nato pretvorili v format mp3, ki je izgubljev (ang. lossy) glasbeni zapis. Mp3 posnetke smo nato pretvorili v posnetke wav (Waveform Audio File Format). Ker, flv in mp3 formata nekatere frekvence izpustita, da bi bila velikost datoteke čim manjša, smo zaradi te metodologije verjetno dosegli slabše rezultate, kot, če bi imeli prvotne posnetke v wav formatu.

**Tabela 6.1:** Posnetki glede na razlog šuma

Šum	Število posnetkov	Trajanje [sek]
smeh dojenčkov	1	180
zvoki psov in televizije	1	145
prometna cesta (avtomobili, hupe, pogovori)	3	960
mestni park (živali, otroško igranje, pogovori odraslih)	3	862
zvoki bele tehnike in čiščenja stanovanja	3	919
pogovori v klicnem centru	1	180
pogovori v kavarni	1	320
pogovor v šestih jezikih	1	46
SKUPAJ	14	3612

## 6.2 Pridobitev značilk

Wav format smo pretvorili v tabelo frekvenc in tako za vsak posnetek (dolžine ene sekunde) dobili tabelo velikosti 44000 frekvenc. S to tabelo smo izračunali MFCC koeficiente, delta koeficiente in delta-delta koeficiente. Vsak posnetek, dolžine ene sekunde, je po pridobitvi značilk vseboval 2587 MFCC značilk.

## 6.3 Priprava okolja na oblaku za učenje na-povednih modelov

Obdelavo posnetkov, pridobitev značilk, izbiro parametrov, binarno klasifikacijo, večrazredno klasifikacijo in klasifikacijo z rekurenčno nevronske mreže smo naredili na virtualnih strojih na oblaku OpenStack.

Za paralelno izvajanje izbire parametrov smo ustvarili tri virtualne stroje. Vsakemu stroju smo določili velikost vira m1.xlarge, ki vsebuje 8 vCPUs

(virtualni procesor), 16GB RAMa in 80GB diska.

Za lažji dostop do virtualnih strojev smo vsakemu stroju dodelili javni IP naslov in virtualnim strojem dodelili varnostno skupino, ki dovoljuje dostop s protokolom SSH (na vratih 22 smo za vse naslove IP dovolili TCP promet proti stroju (ang. Ingress direction)). Za dodatno varnost smo poskrbeli tako, da je povezava na virtualne stroje dovoljena samo z zasebnim ključem. Dostop z geslom je onemogočen.

## 6.4 Binarna klasifikacija

Za binarno klasifikacijo smo uporabili metode logistična regresija, naključni gozdovi, k-najbližjih sosedov in metodo podpornih vektorjev. Klasifikacijsko natančnost smo izračunali s pomočjo deset kratnega prečnega preverjanja, kjer nismo spreminjali osnovnih parametrov uporabljenih metod.

Zaradi visoke klasifikacijske natančnosti in nevarnosti prevelikega prilagajanja modela učnim podatkom smo poskrbeli, da v vseh nadaljnjih testih primeri iz istega posnetka ne nastopajo v učni in testni množici hkrati.

Nadalje smo preverili klasifikacijske natančnosti za osnovne parametre algoritmov in izbrali parametre, ki za naše podatke kažejo najvišjo klasifikacijsko natančnost.

### Validacija nad podatki, ki vsebujejo iste posnetke v učni in testni množici

Prvi korak metodologije zajema klasifikacijo v dva razreda: jok in ne-jok. Za prvi občutek uspešnosti metod za klasifikacijo, smo vse primere razbili na dolžino ene sekunde. S tem smo dobili 6387 posameznih primerov. Vsak primer ima 2587 značilk. S pomočjo deset kratnega prečnega preverjanja smo preverili natančnost uporabljenih metod. Parametrov metod pri tem testu nismo spreminjali. Primerjavo uspešnosti metod vidimo v tabeli 6.2.

**Tabela 6.2:** Primerjava natančnosti metod za binarno klasifikacijo z 10 kratnim prečnim preverjanjem.

Metoda	Logistična regresija	Naključni Gozdovi	KNN	SVM
CA/AUC	0.86/0.86	0.93/0.93	0.91/0.90	0.97/0.97

**Tabela 6.3:** Natančnost binarne klasifikacije za metodo logistične regresije za različne parametre. Deset kratno prečno preverjanje.

Regularizacija	Lasso(L1)				
C	0.1	0.5	1	<b>10</b>	20
CA/AUC	0.81/0.80	0.81/0.80	0.81/0.80	<b>0.82/0.81</b>	0.82/0.81

Regularizacija	Ridge(L2)				
C	0.1	0.5	1	10	20
CA/AUC	0.81/0.8	0.81/0.8	0.81/0.8	0.81/0.8	0.81/0.8

### Validacija nad podatki, ki ne vsebujejo istih posnetkov v učni in testni množici

Zaradi visoke natančnosti posameznih metod smo preverili, če je gradnja modela bila smiselna in ali smo mogoče zaradi razbitja posameznega posnetka na več primerov, ki jih nato obravnavamo samostojno, omogočili, da so naše metode pristrane. Zaradi tega smo v vseh nadaljnjih testih poskrbeli, da v učni množici podatkov in v testni množici podatkov ne nastopajo primeri iz istih posnetkov.

Primerjavo natančnosti za različne parametre metode logistične regresije, lahko vidimo v tabeli 6.3. Primerjavo natančnosti za različne parametre metode naključnih gozdov, lahko vidimo v tabeli 6.4. Primerjavo natančnosti za različne parametre k najbližjih sosedov, lahko vidimo v tabeli 6.5. Primerjavo natančnosti za različne parametre metode podpornih vektorjev, lahko vidimo v tabeli 6.6. Zbrane rezultate natančnosti po izbranih parametrih vidimo v tabeli 6.7.



**Tabela 6.4:** Natančnost binarne klasifikacije za metodo naključnih gozdov glede na število dreves. Deset kratno prečno preverjanje.

Število dreves	5	10	<b>15</b>	20
CA/AUC	0.79/0.80	0.81/0.81	<b>0.83/0.83</b>	0.83/0.83

**Tabela 6.5:** Natančnost binarne klasifikacije za metodo k najbližjih sosedov glede na parametre. Deset kratno prečno preverjanje.

Mera razdalje	Evklidska			Manhattanska		
	5	10	20	<b>5</b>	10	20
CA/AUC	.83/.81	.83/.8	.82/.79	<b>.84/.82</b>	.83/.81	.83/0.8

**Tabela 6.6:** Natančnost binarne klasifikacije za metodo podpornih vektorjev glede na parametre. Deset kratno prečno preverjanje.

Jedro	Radialno			
<b>C</b>	0.01	<b>0.1</b>	0.5	1
CA/AUC	0.84/0.84	<b>0.89/0.89</b>	0.89/0.88	0.88/0.88

Jedro	Linearno			
<b>C</b>	0.01	0.1	0.5	1
CA/AUC	0.82/0.81	0.8/0.79	0.8/0.79	0.8/0.79

Jedro	Sigmoidno			
<b>C</b>	0.01	0.1	0.5	1
CA/AUC	0.81/0.79	0.82/0.8	0.78/0.77	0.76/0.75

**Tabela 6.7:** Združena tabela natančnosti klasifikacije za jok/ne-jok z izbranimi parametri.

Metoda	Logistična regresija	Naključni Gozdovi	KNN	<b>SVM</b>
CA/AUC	0.82/0.81	0.83/0.83	0.84/0.82	<b>0.89/0.89</b>

**Tabela 6.8:** Primerjava natančnosti metod za binarno klasifikacijo z izbranimi parametri in deset kratnim prečnim preverjanjem. Odstranjena tišina v posnetkih.

Metoda	Logistična regresija	Naključni Gozdovi	KNN	SVM
CA/AUC	0.81/0.79	0.84/0.84	0.84/0.82	<b>0.89/0.87</b>

### Klasifikacija z ločenimi otroci v učni in testni množici in z odstranjeno tišino v posnetkih

V tabeli 6.8 vidimo, da dobimo slabše rezultate s posnetki, ki smo jim odstranili tišino. S tem ko smo odstranili tišino smo odstranili vdihe in izdihe otrok, ki so tudi sestavni del joka [69].

## 6.5 Večrazredna klasifikacija

Za večrazredno klasifikacijo smo uporabili metode logistična regresija, naključni gozdovi, k-najbližjih sosedov in metodo podpornih vektorjev. Preverili smo klasifikacijske natančnosti za osnovne parametre algoritmov in izbrali parametre, ki za naše podatke kažejo najvišjo klasifikacijsko natančnost. Pri deset kratnem prečnem preverjanju smo poskrbeli, da v učni in testni množici nimamo posnetkov iz istih posnetkov. Za izbrane parametre smo poročali klasifikacijsko natančnost po razredih, povprečno klasifikacijsko natančnost, preciznost po razredih, povprečno preciznost, priklic po razredih in povprečen priklic.

Klasificirali smo v razrede bolečina, izločanje, jeza, lakota, strah in zanesljivost.

Podrobno analizo osnovnih parametrov in mer uspešnosti vidimo v prilogi D. Zbrane rezultate po izbranih parametrih vidimo v tabeli 6.9.

**Tabela 6.9:** Tabela natančnosti za posamezne razrede, povprečna natančnost, povprečna preciznost in povprečni priklic.

	1	2	3	4	5	6	AC	PRE	REC
A	0.73	0.85	0.71	0.52	0.91	0.81	0.76	0.27	0.26
B	0.75	0.88	0.75	0.5	0.94	0.86	0.78	0.29	0.34
C	0.69	0.82	0.78	0.56	0.92	0.86	0.77	0.32	0.31
D	0.81	0.91	0.83	0.47	0.95	0.88	0.81	0.28	0.42

1: bolečina; 2: izločanje; 3: jeza; 4: lakota; 5: strah; 6: zaspanost.

A: Logistična regresija z L1 regularizacijo in  $C=10$ .

B: Naključni gozdovi s številom dreves 15.

C: K najbližjih sosedov z evklidsko razdaljo. Število sosedov je 5.

D: Metoda podpornih vektorjev s Sigmoid jedrom in  $C=0.01$ .

## 6.6 Rekurenčne nevronske mreže

Za večrazredno klasifikacijo smo uporabili tudi rekurenčne nevronske mreže. Primerjali smo različne strukture nevronske mreže z različno strukturiranimi vhodnimi podatki. Primerjali smo tudi, kako na različno zgradbo nevronskih mrež vpliva, če pri vhodnih podatkih uporabimo značilke delte in dvojne delte ali pa samo MFCC značilke.

Pri deset kratnem prečnem preverjanju smo poskrbeli, da v učni in testni množici nimamo posnetkov iz istih posnetkov. Za izbrane parametre smo poročali klasifikacijsko natančnost po razredih, povprečno klasifikacijsko natančnost, preciznost po razredih, povprečno preciznost, priklic po razredih in povprečen priklic.

Primer zgradbe rekurenčne mreže lahko vidimo v prilogi B. Zbrane rezultate po izbranih parametrih vidimo v tabeli 6.10.

**Tabela 6.10:** Tabela natančnosti za posamezne razrede za klasifikacijo z rekurenčno nevronske mreže, povprečna natančnost, povprečna preciznost in povprečni priklic.

	1	2	3	4	5	6	AC	PRE	REC	L1	L2	B	E
A	0.85	0.91	0.84	0.42	0.95	0.88	0.81	0.18	0.42	50	/	32	2
B	0.79	0.87	0.73	0.42	0.94	0.83	0.76	0.25	0.29	10	/	5	3
C	0.82	0.90	0.81	0.47	0.94	0.83	0.80	0.29	0.38	10	/	5	3
D	0.84	0.91	0.81	0.41	0.95	0.87	0.80	0.28	0.40	10	/	16	4
E	0.78	0.86	0.75	0.51	0.94	0.86	0.78	0.31	0.35	100	/	32	10
F	0.79	0.84	0.79	0.50	0.91	0.82	0.77	0.31	0.32	50	100	32	10

1: bolečina; 2: izločanje; 3: jeza; 4: lakota; 5: strah; 6: zaspanost.

A: Vhodni nivo Embedding. Drugi nivo LSTM s 50 nevroni. Značilke združene.

B: Vhodni nivo LSTM z 10 nevroni. Značilke združene.

C: Vhodni nivo LSTM z 10 nevroni. Značilke ločene.

D: Vhodni nivo LSTM z 10 nevroni in spustom. Značilke ločene.

E: Vhodni nivo LSTM z 10 nevroni in spustom. Značilke ločene.

F: Vhodni nivo LSTM z 10 nevroni, spustom in dodatnim LSTM nivojem. Značilke ločene.

**Tabela 6.11:** Primerjava metrik glede na starost otroka

Starost	0-3	3-6	6-12
AUC/CA	0.72/0.78	0.80/0.70	0.77/0.73

## 6.7 Primerjava po starosti

Na koncu smo posnetke razdelili v tri skupine. Prva skupina vsebuje vse otroke do tretjega meseca starosti, druga skupina vsebuje otroke starosti od treh do šest mesecev starosti in tretja skupine vsebuje starejše otroke od šest mesecev. Z modelom podpornih vektorjev smo izračunali klasifikacijsko natančnost in površino pod krivuljo. Zaradi omejenih podatkov pri tem testu nismo pazili, da primeri iz istih posnetkov niso ločeni na učno in testno množico. Zaradi tega ne moremo rezultatov tega testa upoštevati pri natančnosti našega modela, lahko pa vidimo korelacije in kako se spreminja natančnost glede na starost. Rezultati so prikazani v tabeli 6.11.

S tem rezultatom smo ovrednotili hipotezo te magistrske. Vidimo, da (vsaj na naših podatkih) starost ne vpliva na klasifikacijsko natančnost.



## Poglavje 7

# Diskusija in nadaljnje delo

V magistrskem delu smo raziskovali povezavo, da bo starost dojenčkov vplivala na akustične aspekte joka. To bi se pokazalo v različni klasifikacijski uspešnosti modelov za napoved razreda iz joka. Na podlagi naših rezultatov povezave ne opazimo, kvečjemu jo lahko zavrnamo. Vseeno obstajajo nekateri pomisleki, ki nam preprečujejo samozavestno zavrnitev povezave med starostjo dojenčkov in klasifikacijske uspešnosti. Namreč, format zapisa zvoka je bil zaradi kompresije v flv in mp3 osiromašen v svoji frekvenčni sestavi. Morda bi format wav lahko bil dovolj bogat, da bi vseboval razlike med joki dojenčkov različnih starosti. Poleg tega bi bila raziskava z večjo podatkovno bazo bolj uspešna. Čeprav smo nabrali veliko več primerov od povprečne raziskave, opisane v pregledu sorodnih del, pa naši posnetki niso predstavljali veliko posameznih govorcev v določenih kategorijah joka. To bi lahko zmanjšalo napovedno natančnost starosti, saj prave primerjave niso bile možne med vsemi kategorijami. Kljub neuspehu potrditve povezave smo pokazali izvedljivost aplikacije, ki bi podatke lahko pošiljala na oblak. Slednje omogoča razširljivost aplikacije na številne naprave, ne da bi žrtvovali procesorsko moč ali varnost podatkov pri obdelavi posnetkov. Hkrati smo potrdili, da lahko jok uspešno klasificiramo v razrede joka. Tako smo s pomočjo mere natančnosti klasifikacijskega modela pokazali, da je tišina ali pomanjkanje joka v določenih posnetkih pomembna informacija za klasifikacijo joka. Opo-

zorili smo na potencialno zanko mere točnosti, ki zakrije neuspešnost modela v večrazrednih klasifikacijah. Z upadcem uspešnosti klasifikacijskega modela pri izključujoči delitvi *govorcev* na učno in testno množico smo pokazali, da je uspešnost večine klasifikacijskih primerov verjetno nižja od poročane, saj druge študije (z izjemo ene) ne poročajo o prečnem preverjanju čez govorce, ampak zgolj čez posnetke njihovih jokov. Kot pričakovano iz sistematskega pregleda sorodnih virov literature, sta metoda podpornih vektorjev in nevronske mreže bila najbolj uspešna klasifikatorja joka. V kolikor bomo v prihodnje še dalje raziskovali klasifikacijo dojenčkovih jokov ali njeno izvedljivost preko oblaka, bomo verjetno gradili dalje na teh klasifikatorjih. Čeprav smo vse modele klasifikacije preizkušali in izpopolnjevali s številnimi parametri (podrobnejši opis v prilogi), v model klasifikacije nismo vključili številne pristope, ki bi lahko potencialno izboljšali klasifikacijsko uspešnost. Tu bomo orisali zgolj nekatere možnosti za izboljšave v prihodnjih raziskavah:

- Avtorji Aucouturier in sod. [69] so s skritimi Markovimi modeli uspešno binarno klasificirali dojenčkov jok na fazo vdiha in izdiha, klasifikacija je bila robustna za različne starosti, vokalizacijske kontekste in različne dojenčke. Tako klasifikacijo bi lahko obravnavali v predobdelavi podatkov in potencialno izboljšali klasifikacijsko uspešnost.
- Za pridobivanje značilk smo uporabili MFCC, saj so se v številnih raziskavah izkazali uspešni za klasifikacijske namene. Trenutni razvoj na področju klasifikacije zvočnih signalov pa namiguje, da bi lahko logaritmi amplitud Mel spektrogramov bile boljše značilke, predvsem za skrite nevronske mreže [70, 71].
- Tarčni razredi so morda bili napačno izbrani. Bgnic in sod. [72] so pokazali, da je klasifikacija strokovnjakov za novorojenčke bila velikokrat napačna, saj je znotraj razreda nelagodja vključevala druge razrede lakote in bolečine. Honda in sod. [22] so v predprocesiranju signala ugotovili, da je nesoglasje med klasifikacijo mater in klasifikacijo strokovnjakov mogoče zvesti na gruče, kjer je nesoglasje bilo vedno



med razredi znotraj iste gruče (gruča 1: zaspanost in nejevolja, gruča 2: jeza, žalost in lakota). Obe raziskavi namigujeta, da moramo biti nekoliko skeptični do lastnih oznak klasifikacij v razrede znotraj nadzorovanega učenja. Močnejša študija, ki nenadzorovano določi kategorije med dojenčkovimi joki bi lahko bila izredno informativna.

- V področju raziskav dojenčkovih jokov moramo bolj dosledno uporabljati mere uspešnosti klasifikacijskih modelov. Kot smo pokazali, je točnost lahko zavajajoča mera, ki ne sme biti uporabljena izven konteksta ali drugih mer.
- Namesto MFCC bi lahko izbrali za pridobivanje značilk tako LPCC kot PLP, ker sta se izkazala za koristna v številnih sistemih avtomatične klasifikacije zvoka [42, 73].

Pokazali smo uporabnost oblačnih storitev za hrambo in obdelavo podatkov. Zaključujemo raziskovalno delo z usmeritvami za prihodnje, od metodoloških popravkov pa do boljših tehnik pridobivanja značilk za klasifikatorje.



# Dodatek A

## Jok

Ljudje uporabljamo zvok kot sredstvo sporazumevanja in kot orodje osebnega izražanja. Zvok uporabljamo za izražanje svojega prepričanja, petja, oponašanja drugih zvokov, umetniška izražanja in celo za izražanje čustvenega stanja (kričanje, smeh in jok). Zvok je eden izmed načinov, ki nam omogoča, da lahko svoje želje, počutje in druga notranja stanja izrazimo v zunanjem okolju.

Dojenčki, ki še ne znajo govoriti, so pri taki komunikaciji veliko bolj omejeni kot pa odrasli. Njihov zvočni komunikacijski sistem bi lahko poenostavljeno razumeli v povsem valenčni strukturi glede na izrazna orodja, ki so jim na voljo: smeh kot pozitivni odziv, jok kot negativni odziv in odsotnost smeha ali joka kot nevtralni odziv.

Vendar je tako razumevanje dojenčkovega zvočnega komuniciranja prevelika poenostavitev in je notranja delitev občutij bolj kompleksna. Dojenček lahko joče zaradi lakote, bolečine, vročinskega stanja, strahu ali drugih razlogov. V takih primerih je tako za starše ali negovalce kot morda za zdravstveno osebje v primeru bolezenskih stanj ključno razumeti razloge za jok, da se lahko na njih pravilno odzovemo. Neprimerno odpravljanje vzrokov (npr. previjanje dojenčka namesto hranjenja, ko se le-ta joče) je lahko stresno za negovalce in otroke, hkrati pa je v ekstremnih primerih celo nevarno. Tako bi na primer napačno razumevanje joka s strani zdravstvenega osebja

lahko zavlačevalo okrevanje bolnega dojenčka ali pa celo zanemarilo vzrok za pravo bolezensko stanje.

Strukturo joka lahko razumemo povsem avditivno, torej glede na zvočni signal, čigar analiza omogoča klasifikacijo. Uspešen model, ki bi razbral kategorični vzrok, bi lahko hkrati tudi napovedal vzrok na primeru novega, tj. še ne-analiziranega signala.

## A.1 Lastnosti zvočnega signala pri joku

Ljudje proizvajamo zvoke govora, joka, smeha in petja skozi isti zvočni organ, ki producira zvočni signal. Zvočni signal vedno proizvaja isti biološki aparat, a se signal razlikuje med različnimi razlogi proizvodnje zvoka - smeh in jok zvenita in sta proizvedena drugače.

### A.1.1 Biološki aparat, ki proizvede zvočni signal

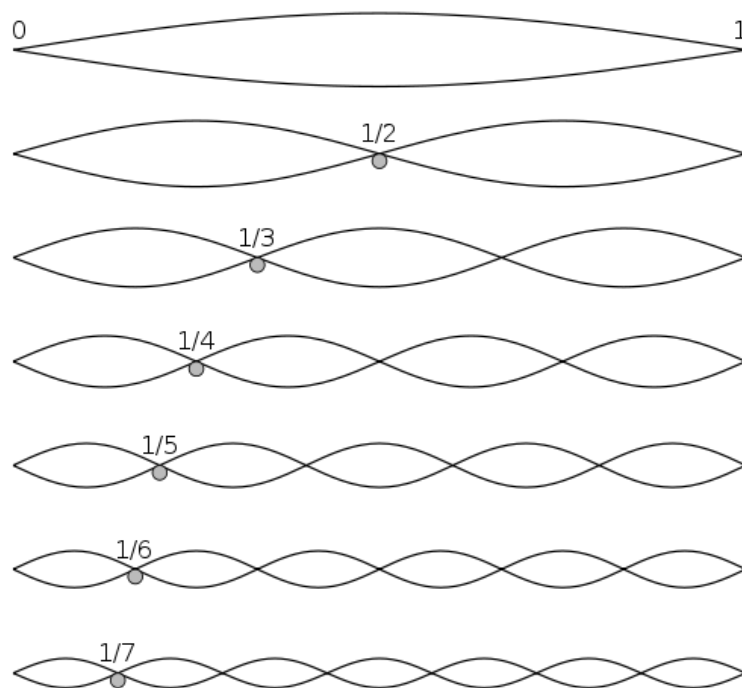
Zvočni signal nastane s pomočjo našega biološkega zvočnega organa. Zvočni organ je v širšem razumljen kot tridelni sistem, sestavljen iz pljuč, glasilk in vokalnega trakta [74]. Sundberg [74, str. 10] poda analogijo teh komponent skozi njihovo mehanično funkcijo: pljuča kot kompresor zraka, glasilke kot nihalo in vokalni trakt kot rezonator. Pljuča delujejo kot kompresor zraka, ki pod pritiskom potisnejo zračni tok iz pljuč preko sapnice skozi glasilno špranjo (glotis) med glasilkami. Zračni tok je potisnjen skozi glasilno špranjo pod pritiskom, ta pritisk pa proizvede intervalno odpiranje in zapiranje glasilne špranje. Odpiranje in zapiranje povzroči zgoščine in razredčine v zračnem toku v obliki intervalnih paketov zračnega toka. Hitrost odpiranja in zapiranja glasilk je enaka frekvenci zračnega toka, ki ga sestavljajo molekule zgoščene v zgoščinah. Te iste molekule so potisnjene dalje v vokalni trakt, ki jih dodatno zaniha zaradi svoje oblike in lastne vibracije. Vokalni trakt, vključujoč spodnji del grla in zgornji ustno-nosni del, rezonira vibracije zraka kot filter — lahko jih ojača ali oslabi.

### A.1.2 Fizikalne lastnosti zvočnega signala so odvisne od biološkega aparata

Zračni tok zapusti zvočni organ v obliki mehanskega valovanja molekul, ki se širijo po okolici. Valovanje je tukaj razumljeno kot intervalno menjavanje zgoščin in razredčin molekul v zračnem toku, ki jih lahko opišemo s sinusno funkcijo. Ko je zvočni signal v okoliškem mediju, se zračni tok razlikuje glede na njegove fizikalne prvine, ki ga je proizvedel biološki zvočni organ. Te prvine so oblika zračnega toka, razdalja med razredčinami in zgoščinami molekul v toku (valovna dolžina in sorodno izpeljana frekvenca) ter pritisk (amplituda), s katerim zračni tok zapusti zvočni organ. Jakost oziroma pritisk izdihavanja, koti pod katerim je zračni tok bil potisnjen skozi glasilno špranjo, spreminjanje pritiska toka ali njegove hitrosti znotraj istega izdihavanja zraka, hitrost nihanja glasilk (tj., njihovega odpiranja in zapiranja), število nihajočih glasilk, morfološka oblika vokalnega trakta in spreminjanje oblike vokalnega trakta med izdihavanjem, so zgolj nekatere lastnosti biološkega aparata, ki vplivajo na fizikalne lastnosti zvočnega signala.

Za razumevanje človeškega zvočnega signala moramo posebej obravnavati nihanje, ki ga povzročajo glasilke z odpiranjem in zapiranjem glasilne špranje. Hitrost odpiranja in zapiranja glasilne špranje je enakovredna frekvenci zvočnega toka, ki pride skozi. Torej pri frekvenci 440 Hz se glasilke odprejo in zaprejo 440-krat v eni sekundi. Frekvenci sta enaki, saj je ravno zapiranje in odpiranje glasilk, tisto, ki povzroča zgoščine in razredčine molekul v zračnem toku, ki gre skozi glasilke. Čas med eno zgoščino in naslednjo je pa čas od zadnjega odprtja preko zaprtja do naslednjega odprtja. Čas med dvema zgoščinama je hkrati frekvenca zvočnega toka.

Zgoščine in razredčine molekul v zračnem toku lahko opišemo s sinusoido, kjer je razdalja med dvema sosednjimi maksimumi (ali minimumi) valovna dolžina. Frekvenca je število ciklov (med sosednjim minimumom in maksimumom), ki jih zvočni tok opravi v eni sekundi. Amplituda zvočnega signala ali pritisk zračnega toka je odražena v amplitudi sinusoide oziroma maksimalni razdalji med srednjo vrednostjo sinusoide in njenim maksimumom ali



**Slika A.1:** Harmonične frekvence idealnega nihala [75].

minimumom.

Najnižja frekvenca, ki jo glasilke proizvedejo, se imenuje osnovna frekvenca ( $f_0$ ). Če glasilke nihajo hitreje, ustvarijo harmonične frekvence, ki so večkratniki osnovne frekvence. Prva harmonična frekvenca je torej 2x hitrejša od osnovne, druga harmonična 3x hitrejša in tako dalje. Slika A.1 prikazuje kako v idealiziranem tonu poteka osnovna frekvenca in njene harmonične frekvence na primeru matematično idealiziranega nihala. Na sliki pika označuje dolžino nihala od levega roba do desne skrajnosti nihala. Vidimo, da je frekvenca povezana z dolžino nihala. Matematično idealizirano nihalo s polovično dolžino osnovnega nihala proizvaja dvakratno frekvenco osnovnega nihala, tretjina dolžine osnovnega nihala proizvaja trikrat višjo frekvenco itd. kot vidimo na sliki A.1.

Glasilke delujejo podobno kot matematično nihalo: glede na to kolikšen delež njihove dolžine niha, glasilke proizvajajo večkratnike frekvenc osnovne

frekvence. Te večkratnike imenujemo harmonične frekvence tudi v primeru človeškega zvočnega signala. Ker so osnovna frekvenca in njene izpeljane harmonične frekvence odvisne od dolžine nihala (oziroma glasilk v primeru človeškega zvočnega signala), obstajajo razlike v osnovni frekvenci med ljudmi. Otroci imajo krajše glasilke od odraslih in zato so njihove osnovne frekvence ponavadi višje ter se manjšajo s starostjo [74, str. 102]. Podobno velja za razlike med odraslimi posamezniki: vsak posameznik ima morfološko edinstvene glasilke in posledično nekoliko drugačno osnovno frekvenco od drugih posameznikov [74, str. 102].

Kot že opisano, se frekvenca zvočnega signala spremeni v rezonatorju oziroma vokalnem traktu. Zaradi lastne oblike in nihanja spremeni osnovno frekvenco in njene izpeljane harmonične frekvence v nove frekvence imenovane formanti. Vokalni trakt večinoma ojača samo osnovne in harmonične frekvence, medtem ko zvoke drugih frekvenc ošibi. Tako je prvi formant F1 nekoliko spremenjena osnovna frekvenca, drugi formant F2 je spremenjena prva harmonična frekvenca, itd. To je pomembno za analizo, saj je fizikalna lastnost zvočnega signala, ki pride iz govora ali joka, amplitudno ojačana formantna frekvenca, ki je nekoliko spremenjena osnovna frekvenca ali harmonična frekvenca.

Seveda med proizvajanjem zvočnega signala govora, smeha, joka ali drugega zvočnega signala proizvajamo zvoke, ki imajo še druge lastnosti poleg amplitude in frekvence. Tako so pomembni dejavniki zvočnega signala šum, nezveneči zvoki (npr. šepeti, kliki, piskanje, itd.), pavze, vdih in podobno. A formantne frekvence so ključna prvina, saj kažejo neko konsistenco oziroma vzorec, ker so osnovane na nespreminjajoči se osnovni frekvenci in njenih harmoničnih frekvencah.

## A.2 Individualne in skupinske razlike v proizvodjanju zvokov med jokom

Prvine kot so osnovna frekvenca in formantne frekvence so dokaj nevariabilne za istega govorca. Obstajajo pa velike razlike med različnimi posamezniki in skupinami posameznikov.

Razlike med posamezniki v akustični sliki proizvedenih zvokov se pojavijo zaradi morfologije oziroma zgradbe in sestave govornega aparata vsakega govorca, ki daje govorcju tako med jokom kot pri proizvodjanju drugih zvokov svoj edinstven pečat [74].

Skupine posameznikov pa kažejo skupne prvine, zaradi katerih lahko ločimo skupine med seboj. Moški imajo ponavadi nižje osnovne in formantne frekvence od žensk, slednje pa nižje od otrok. Osnovna in formantne frekvence so odvisne od dolžine glasilk, saj daljše glasilke proizvajajo nižje frekvence [76]. Glasilke odraslega moškega so v povprečju dolge med 15 in 20 mm, povprečne odrasle ženske glasilke so v razponu med 9 in 12 mm, novorojenčkove pa okoli 3mm [74, str. 6]. Poleg spola, starost vpliva na velikost glasilk in nižino frekvenc, ki jih proizvajajo. Zato imajo starejši otroci ponavadi nižje formantne frekvence od mlajših [74, str. 102-110]. Ni pa povsem jasno, ob kateri starosti in v kakšni meri nastopijo razlike tako med spoloma kot med posamezniki različnih starosti. Pri dveh mesecih ni še razlike med spoloma v formantih frekvencah [77]. Nasprotno, Daga in Panditrao [78] poročata o različnih osnovnih frekvencah med novorojenčki različnih spolov, medtem ko drugi avtorji poročajo, da so frekvenčne razlike med spoloma pri dojenčkih premajhne v primerjavi s frekvenčnimi razlikami med joki z različnimi razlogi za jok [8].

Čeprav ima spol vprašljiv vpliv na akustiko joka, starost vpliva na zvok joka na še dodaten, psihološko-socialen način. Dojenčki različnih starosti se jočejo na kvalitativno drugačne načine. Boukydis in Lester [77, str. 18-19] opišeta kako razvojna stopnja vpliva na otroke, saj dojenčki do konca prvega ali drugega meseca jočejo v odzivu na fiziološke stimulacije (npr. bolečina,



mraz, lakota), kasneje pa začnejo razvijati še bolj kompleksne kognitivno-čustvene razloge za jok, kot na primer jok zaradi iskanja pozornosti, zaradi jeze, jok zaradi strahu pred neznanci in ločitvijo od starša in podobno.

Na podlagi individualnih in skupinskih razlik lahko sklepamo, da bo starost vplivala na klasifikacijski model in ugibamo, da bi spol morda lahko tudi vplival na model.



## **Dodatek B**

# **Primerjava mer uspešnosti za večrazredno klasifikacijo**

Primerjavo uspešnosti glede na izbrane parametre lahko vidimo na sliki B.1.

DODATEK B. PRIMERJAVA MER USPEŠNOSTI ZA VEČRAZREDNO  
62 KLASIFIKACIJO

	BOLEČINA			IZLOČANJE			JEZA			LAKOTA			STRAH			ZASPANOST			POVPREČJE		
	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall
C	<b>Lasso L1</b>																				
0.1	0.73	0.18	0.22	0.84	0.08	0.07	0.71	0.17	0.19	0.53	0.44	0.4	0.91	0.13	0.11	0.81	0.21	0.21	0.76	0.28	0.27
0.5	0.74	0.18	0.21	0.84	0.09	0.08	0.7	0.16	0.18	0.52	0.43	0.39	0.91	0.11	0.1	0.81	0.22	0.22	0.75	0.27	0.26
1	0.74	0.18	0.2	0.84	0.09	0.08	0.71	0.16	0.19	0.52	0.43	0.39	0.91	0.13	0.12	0.81	0.22	0.22	0.76	0.27	0.27
10	0.73	0.16	0.19	0.85	0.08	0.07	0.71	0.16	0.18	0.52	0.42	0.38	0.91	0.14	0.14	0.81	0.21	0.21	0.76	0.27	0.26
20	0.73	0.16	0.18	0.85	0.09	0.07	0.71	0.16	0.18	0.52	0.42	0.39	0.91	0.15	0.14	0.81	0.21	0.22	0.76	0.27	0.26
C	<b>Ridge L2</b>																				
0.1	0.74	0.19	0.22	0.85	0.07	0.05	0.71	0.16	0.18	0.53	0.43	0.4	0.91	0.14	0.12	0.81	0.2	0.2	0.76	0.27	0.27
0.5	0.74	0.18	0.22	0.85	0.07	0.06	0.71	0.16	0.18	0.52	0.43	0.4	0.91	0.13	0.12	0.8	0.2	0.2	0.76	0.27	0.27
1	0.74	0.19	0.22	0.85	0.07	0.06	0.71	0.16	0.18	0.52	0.43	0.4	0.91	0.13	0.12	0.8	0.2	0.2	0.76	0.27	0.27
10	0.74	0.18	0.21	0.85	0.08	0.07	0.72	0.16	0.18	0.52	0.42	0.41	0.91	0.12	0.12	0.8	0.2	0.2	0.76	0.27	0.27
20	0.74	0.18	0.21	0.85	0.07	0.06	0.71	0.16	0.18	0.52	0.42	0.41	0.91	0.13	0.12	0.8	0.2	0.2	0.76	0.27	0.27
Št. dreves	<b>Random Forests</b>																				
5	0.7	0.16	0.23	0.86	0.1	0.06	0.72	0.17	0.19	0.52	0.43	0.45	0.92	0.05	0.03	0.83	0.21	0.14	0.76	0.27	0.28
10	0.71	0.13	0.16	0.87	0.12	0.07	0.76	0.24	0.21	0.49	0.42	0.54	0.94	0.15	0.03	0.84	0.2	0.1	0.77	0.28	0.3
15	0.75	0.15	0.15	0.88	0.08	0.03	0.75	0.23	0.23	0.5	0.43	0.62	0.94	0.06	0.01	0.86	0.31	0.12	0.78	0.29	0.34
20	0.75	0.12	0.1	0.88	0.09	0.04	0.76	0.27	0.28	0.5	0.43	0.61	0.94	0.11	0.02	0.85	0.28	0.13	0.78	0.29	0.34
Št. sosedov	<b>k-NN Euklidian</b>																				
5	0.69	0.1	0.14	0.82	0.09	0.11	0.78	0.34	0.33	0.56	0.47	0.5	0.92	0.07	0.04	0.86	0.35	0.14	0.77	0.32	0.31
10	0.68	0.09	0.12	0.81	0.06	0.08	0.78	0.32	0.31	0.52	0.44	0.49	0.93	0.02	0.01	0.87	0.34	0.1	0.77	0.3	0.29
20	0.68	0.07	0.09	0.8	0.07	0.1	0.78	0.32	0.3	0.5	0.42	0.49	0.94	0.03	0.01	0.87	0.37	0.08	0.76	0.29	0.29
Št. sosedov	<b>k-NN Manhattan</b>																				
5	0.7	0.15	0.2	0.81	0.08	0.1	0.79	0.33	0.3	0.53	0.45	0.48	0.92	0.11	0.06	0.86	0.29	0.11	0.77	0.31	0.3
10	0.69	0.09	0.12	0.8	0.06	0.09	0.78	0.31	0.27	0.51	0.42	0.49	0.93	0.09	0.03	0.87	0.29	0.07	0.76	0.29	0.29
20	0.69	0.08	0.1	0.79	0.06	0.08	0.77	0.26	0.21	0.47	0.4	0.5	0.94	0.04	0.01	0.87	0.28	0.05	0.76	0.26	0.27
C	<b>SVM RBF</b>																				
0.01	0.85	0	0	0.91	0	0	0.84	0	0	0.42	0.42	1	0.95	0	0	0.88	0	0	0.81	0.18	0.42
0.1	0.85	0	0	0.91	0	0	0.84	0	0	0.42	0.42	1	0.95	0	0	0.88	0	0	0.81	0.18	0.42
0.5	0.85	0	0	0.91	0	0	0.84	0	0	0.42	0.42	1	0.95	0	0	0.88	0	0	0.81	0.18	0.42
1	0.85	0	0	0.91	0	0	0.84	0	0	0.42	0.42	1	0.95	0	0	0.88	0	0	0.81	0.18	0.42
C	<b>SVM Linear</b>																				
0.01	0.72	0.22	0.33	0.83	0.13	0.15	0.74	0.22	0.23	0.55	0.45	0.36	0.91	0.13	0.12	0.83	0.28	0.26	0.76	0.31	0.29
0.1	0.72	0.22	0.33	0.83	0.13	0.15	0.74	0.22	0.23	0.55	0.45	0.36	0.91	0.13	0.12	0.83	0.28	0.26	0.76	0.31	0.29
0.5	0.72	0.22	0.33	0.83	0.13	0.15	0.74	0.22	0.23	0.55	0.45	0.36	0.91	0.13	0.12	0.83	0.28	0.26	0.76	0.31	0.29
1	0.72	0.22	0.33	0.83	0.13	0.15	0.74	0.22	0.23	0.55	0.45	0.36	0.91	0.13	0.12	0.83	0.28	0.26	0.76	0.31	0.29
C	<b>SVM Sigmoid</b>																				
0.01	0.81	0.08	0.03	0.91	0.25	0	0.83	0.39	0.08	0.47	0.44	0.96	0.95	0	0	0.88	0	0	0.81	0.28	0.42
0.1	0.75	0.13	0.12	0.89	0.26	0.12	0.84	0.54	0.19	0.47	0.42	0.71	0.94	0	0	0.83	0.06	0.03	0.79	0.32	0.36
0.5	0.81	0.06	0.02	0.88	0.24	0.14	0.83	0.46	0.22	0.48	0.42	0.68	0.94	0	0	0.76	0.08	0.09	0.78	0.29	0.35
1	0.81	0.06	0.02	0.89	0.32	0.2	0.82	0.41	0.22	0.48	0.43	0.68	0.94	0	0	0.76	0.08	0.09	0.78	0.29	0.35

Slika B.1: Primerjava mer uspešnosti za večrazredno klasifikacijo

## Dodatek C

# Podroben opis posameznih posnetkov joka iz Youtube-a

V spodnji tabeli so razdelani vsi posnetki, ki smo jih pridobili z Youtube-a. Vsaka vrstica je svoj unikatni posnetek, z unikatno številko, starostjo dojenčka v dnevih, starostjo v treh mesečnih intervalih, razredom razloga za jok, spolom dojenčka in dolžino trajanja posnetka.

**Tabela C.1:** Podroben opis zajetih posnetkov

#	Starost [dnevi]	Starost [meseči]	Razlog za jok	Spol	Dolžina [s]
1	30	0-3	lakota	ženski	78
2	30	0-3	lakota	ženski	24
3	7	0-3	lakota	neznano	7
4	8	0-3	izločanje	moški	52
5	8	0-3	jeza	moški	68
6	13	0-3	izločanje	moški	76
7	120	3-6	lakota	neznano	25
8	90	3-6	lakota	ženski	71
9	7	0-3	lakota	ženski	46
10	21	0-3	lakota	moški	17

*DODATEK C. PODROBEN OPIS POSAMEZNIH POSNETKOV JOKA  
64 IZ YOUTUBE-A*

---

11	30	0-3	lakota	ženski	124
12	60	0-3	lakota	moški	189
13	18	0-3	lakota	moški	24
14	12	0-3	lakota	ženski	99
15	210	6-12	jeza	ženski	71
16	270	6-12	jeza	moški	23
17	150	3-6	jeza	ženski	25
18	150	3-6	jeza	ženski	28
19	270	6-12	jeza	moški	21
20	270	6-12	jeza	moški	5
21	270	6-12	jeza	moški	10
22	252	6-12	zaspanost	moški	6
23	210	6-12	jeza	moški	41
24	120	3-6	bolečina	moški	9
25	120	3-6	bolečina	moški	8
26	120	3-6	bolečina	moški	13
27	60	0-3	bolečina	neznano	14
28	60	0-3	bolečina	neznano	16
29	60	0-3	bolečina	moški	4
30	120	3-6	bolečina	moški	4
31	120	3-6	bolečina	moški	8
32	150	3-6	bolečina	moški	8
33	60	0-3	bolečina	ženski	4
34	90	3-6	zaspanost	moški	33
35	30	0-3	zaspanost	ženski	40
36	120	3-6	bolečina	ženski	46
37	60	0-3	lakota	ženski	54
38	1	0-3	lakota	moški	48
39	3	0-3	lakota	moški	50
40	172	3-6	jeza	moški	24
41	172	3-6	jeza	moški	19

---

42	172	3-6	jeza	moški	40
43	5	0-3	jeza	moški	71
44	18	0-3	lakota	ženski	24
45	21	0-3	lakota	ženski	132
46	21	0-3	lakota	ženski	58
47	30	0-3	lakota	moški	36
48	365	6-12	bolečina	moški	16
49	30	0-3	bolečina	ženski	9
50	180	6-12	bolečina	ženski	6
51	120	3-6	bolečina	ženski	6
52	60	0-3	strah	neznano	11
53	60	0-3	strah	moški	7
54	175	3-6	zaspanost	moški	36
55	130	3-6	zaspanost	moški	14
56	42	0-3	zaspanost	ženski	13
57	160	3-6	zaspanost	ženski	49
58	126	3-6	zaspanost	moški	12
59	126	3-6	zaspanost	moški	15
60	126	3-6	zaspanost	moški	6
61	70	0-3	zaspanost	moški	4
62	60	0-3	zaspanost	moški	15
63	60	0-3	zaspanost	moški	7
64	90	3-6	zaspanost	moški	19
65	60	0-3	zaspanost	ženski	18
66	60	0-3	zaspanost	ženski	6
67	90	3-6	zaspanost	ženski	54
68	150	3-6	strah	ženski	4
69	270	6-12	strah	ženski	4
70	270	6-12	strah	ženski	5
71	270	6-12	strah	ženski	4
72	14	0-3	strah	neznano	3

*DODATEK C. PODROBEN OPIS POSAMEZNIH POSNETKOV JOKA  
66 IZ YOUTUBE-A*

---

73	360	6-12	strah	moški	3
74	360	6-12	strah	moški	3
75	180	6-12	strah	moški	4
76	125	3-6	strah	moški	2
77	210	6-12	strah	moški	4
78	210	6-12	strah	moški	6
79	210	6-12	strah	moški	4
80	180	6-12	strah	ženski	6
81	180	6-12	strah	ženski	4
82	60	0-3	strah	moški	10
83	60	0-3	strah	moški	5
84	150	3-6	strah	ženski	7
85	150	3-6	strah	moški	3
86	139	3-6	strah	moški	7
87	139	3-6	strah	moški	5
88	139	3-6	strah	moški	5
89	60	0-3	strah	moški	3
90	360	6-12	strah	ženski	5
91	60	0-3	strah	moški	6
92	285	6-12	strah	moški	4
93	285	6-12	strah	moški	6
94	360	6-12	strah	ženski	17
95	330	6-12	strah	ženski	9
96	90	3-6	strah	moški	2
97	150	3-6	izločanje	moški	2
98	105	3-6	izločanje	moški	60
99	77	0-3	izločanje	neznano	22
100	60	0-3	izločanje	moški	2
101	14	0-3	izločanje	ženski	6
102	10	0-3	izločanje	ženski	12
103	30	0-3	izločanje	moški	14



104	30	0-3	izločanje	moški	4
105	116	3-6	bolečina	moški	15
106	90	3-6	bolečina	ženski	16
107	180	6-12	bolečina	ženski	56
108	120	3-6	bolečina	ženski	11
109	180	6-12	bolečina	ženski	50
110	165	3-6	bolečina	moški	59
111	375	6-12	bolečina	ženski	26
112	90	3-6	bolečina	moški	12



# Dodatek D

## Primer nevronske mreže

Primer zgradbe in metrik uspešnosti v enem prečnem preverjanje.

```
-----  
Layer (type)              Output Shape      Param # Connected to  
=====
```

lstm_3 (LSTM)	(None, 10)	960	lstm_input_3[0][0]
-----			
dropout_3 (Dropout)	(None, 10)	0	lstm_3[0][0]
-----			
dense_3 (Dense)	(None, 6)	66	dropout_3[0][0]
-----			
activation_3 (Activation)	(None, 6)	0	dense_3[0][0]
=====			

Total params: 1,026  
Trainable params: 1,026  
Non-trainable params: 0

```
-----  
None  
Train on 2597 samples, validate on 190 samples  
Epoch 1/4  
2597/2597 [=====] - 14s - loss: 1.5794 - acc: 0.3997 - val_acc: 0.5263  
Epoch 2/4
```

2597/2597 [=====] - 13s - loss: 1.5358 - acc: 0.4066 - val\_acc: 0.5789

Epoch 3/4

2597/2597 [=====] - 13s - loss: 1.5169 - acc: 0.4085 - val\_acc: 0.5895

Epoch 4/4

2597/2597 [=====] - 13s - loss: 1.5188 - acc: 0.4174 - val\_acc: 0.5316

190/190 [=====] - 0s

# Literatura

- [1] T. Fuhr, H. Reetz, C. Wegener, Comparison of supervised-learning models for infant cry classification/vergleich von klassifikationsmodellen zur säuglingsschreianalyse, *International Journal of Health Professions* 2 (1) (2015) 4–15.
- [2] T. Etz, H. Reetz, C. Wegener, A classification model for infant cries with hearing impairment and unilateral cleft lip and palate, *Folia Phoniatica et Logopaedica* 64 (5) (2012) 254–261.
- [3] M. Hariharan, L. S. Chee, S. Yaacob, Analysis of infant cry through weighted linear prediction cepstral coefficients and probabilistic neural network, *Journal of medical systems* 36 (3) (2012) 1309–1315.
- [4] A. Zabidi, L. Khuan, W. Mansor, I. Yassin, R. Sahak, Binary particle swarm optimization for feature selection in detection of infants with hypothyroidism, in: *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2011, pp. 2772–2775.
- [5] R. Sahak, W. Mansor, Y. Lee, A. Yassin, A. Zabidi, Performance of combined support vector machine and principal component analysis in recognizing infant cry with asphyxia, in: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, IEEE, 2010, pp. 6292–6295.

- 
- [6] S. Orlandi, C. A. R. Garcia, A. Bandini, G. Donzelli, C. Manfredi, Application of pattern recognition techniques to the classification of full-term and preterm infant cry, *Journal of Voice* 30 (6) (2016) 656 – 663.
- [7] J. Saraswathy, M. Hariharan, S. Yaacob, W. Khairunizam, Automatic classification of infant cry: A review, in: *2012 International Conference on Biomedical Engineering (ICoBE)*, 2012, pp. 543–548. doi:10.1109/ICoBE.2012.6179077.
- [8] B. F. Fuller, M. R. Keefe, M. Curtin, B. J. Garvin, Acoustic analysis of cries from "normal" and "irritable" infants, *Western journal of nursing research* 16 (3) (1994) 243–253.
- [9] M. Petroni, A. Malowany, C. Johnston, B. Stevens, On the use of artificial neural networks (anns) for the classification of three types of infant cries, in: *Communications, Computers, and Signal Processing*, 1995. Proceedings., *IEEE Pacific Rim Conference on*, IEEE, 1995, pp. 501–504.
- [10] M. Petroni, A. Malowany, C. Johnston, B. Stevens, A comparison of neural network architectures for the classification of three types of infant cry vocalizations, in: *Engineering in Medicine and Biology Society*, 1995., *IEEE 17th Annual Conference*, Vol. 1, IEEE, 1995, pp. 821–822.
- [11] J. Orozco Garca, C. Reyes-García, Clasificación de llanto del bebé utilizando una red neural de gradiente conjugado escalado (2002).
- [12] H. Baeck, M. Souza, A bayesian classifier for baby's cry in pain and non-pain contexts, in: *Engineering in Medicine and Biology Society*, 2003. Proceedings of the 25th Annual International Conference of the IEEE, Vol. 3, IEEE, 2003, pp. 2944–2946.
- [13] S. E. Barajas, C. A. Reyes, Your fuzzy relational neural network parameters optimization with a genetic algorithm, in: *The 14th IEEE International Conference on Fuzzy Systems*, 2005. FUZZ'05, 2005.

- 
- [14] S. E. Barajas-Montiel, C. A. Reyes-Garcia, Identifying pain and hunger in infant cry with classifiers ensembles, in: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vol. 2, IEEE, 2005, pp. 770–775.
- [15] S. E. Barajas-Montiel, C. A. Reyes-García, Fuzzy Support Vector Machines for Automatic Infant Cry Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 876–881. doi:10.1007/978-3-540-37258-5\_107.  
URL [http://dx.doi.org/10.1007/978-3-540-37258-5\\_107](http://dx.doi.org/10.1007/978-3-540-37258-5_107)
- [16] Y. Mima, K. Arakawa, Cause estimation of younger babies' cries from the frequency analyses of the voice-classification of hunger, sleepiness, and discomfort, in: 2006 International Symposium on Intelligent Signal Processing and Communications, IEEE, 2006, pp. 29–32.
- [17] P. Pal, A. N. Iyer, R. E. Yantorno, Emotion detection from infant facial expressions and cries, in: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Vol. 2, IEEE, 2006, pp. II–II.
- [18] Y. Abdulaziz, S. M. S. Ahmad, Infant cry recognition system: A comparison of system performance based on mel frequency and linear prediction cepstral coefficients, in: Information Retrieval & Knowledge Management,(CAMP), 2010 International Conference on, IEEE, 2010, pp. 260–263.
- [19] Y. Abdulaziz, S. M. S. Ahmad, An accurate infant cry classification system based on continuous hidden markov model, in: 2010 International Symposium on Information Technology, Vol. 3, IEEE, 2010, pp. 1648–1652.

- 
- [20] K. Kuo, Feature extraction and recognition of infant cries, in: *Electro/Information Technology (EIT), 2010 IEEE International Conference on*, IEEE, 2010, pp. 1–5.
- [21] A. Rosales-Pérez, C. A. Reyes-García, P. Gómez-Gil, Genetic fuzzy relational neural network for infant cry classification, in: *Mexican Conference on Pattern Recognition*, Springer, 2011, pp. 288–296.
- [22] K. Honda, K. Kitahara, S. Matsunaga, M. Yamashita, K. Shinohara, Emotion classification of infant cries with consideration for local and global features, in: *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, 2012, pp. 1–4.
- [23] S. F. Molaezadeh, M. Salarian, et al., Type-2 fuzzy pattern matching for classifying hunger and pain cries of healthy full-term infants, in: *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, IEEE, 2012, pp. 233–237.
- [24] A. Rosales-Pérez, C. A. Reyes-García, J. A. Gonzalez, E. Arch-Tirado, Infant cry classification using genetic selection of a fuzzy model, in: *Iberoamerican Congress on Pattern Recognition*, Springer, 2012, pp. 212–219.
- [25] R. R. Vempada, B. S. A. Kumar, K. S. Rao, Characterization of infant cries using spectral and prosodic features, in: *Communications (NCC), 2012 National Conference on*, IEEE, 2012, pp. 1–5.
- [26] L. Liu, K. Kuo, S. M. Kuo, Infant cry classification integrated anc system for infant incubators, in: *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, IEEE, 2013, pp. 383–387.
- [27] A. K. Singh, J. Mukhopadhyay, K. S. Rao, Classification of infant cries using epoch and spectral features, in: *Communications (NCC), 2013 National Conference on*, IEEE, 2013, pp. 1–5.



- 
- [28] K. Srihiranon, N. Eiamkanitchat, Application of neuro-fuzzy approaches to recognition and classification of infant cry, in: TENCON 2014-2014 IEEE Region 10 Conference, IEEE, 2014, pp. 1–6.
- [29] S. Bano, K. RaviKumar, Decoding baby talk: A novel approach for normal infant cry signal classification, in: Soft-Computing and Networks Security (ICSNS), 2015 International Conference on, IEEE, 2015, pp. 1–4.
- [30] S. Ntalampiras, I. Potamitis, Canonical correlation analysis for classifying baby crying sound events.
- [31] M. V. V. Bhagatpatil, V. Sardar, An automatic infant’s cry detection using linear frequency cepstrum coefficients (lfcc), International Journal of Scientific & Engineering Research 5 (2014) 12.
- [32] I.-A. Bănică, H. Cucu, A. Buzo, D. Burileanu, C. Burileanu, Automatic methods for infant cry classification, in: Communications (COMM), 2016 International Conference on, IEEE, 2016, pp. 51–54.
- [33] C.-Y. Chang, J.-J. Li, Application of deep learning for recognizing infant cries, in: Consumer Electronics-Taiwan (ICCE-TW), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–2.
- [34] N. Wahid, P. Saad, M. Hariharan, Automatic infant cry pattern classification for a multiclass problem, Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 8 (9) (2016) 45–52.
- [35] X. Huang, A. Acero, H.-W. Hon, R. Foreword By-Reddy, Spoken language processing: A guide to theory, algorithm, and system development, Prentice Hall, 2001.
- [36] F. J. Valverde-Albacete, C. Peláez-Moreno, 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox, PLoS One 9 (1) (2014) e84217.

- [37] D. Lederman, Automatic classification of infants' cries, Citeseer, 2002.
- [38] D. Lederman, A. Cohen, E. Zmora, K. Wermke, S. Hauschildt, A. Stellzig-Eisenhauer, On the use of hidden markov models in infants' cry classification, in: Electrical and Electronics Engineers in Israel, 2002. The 22nd Convention of, IEEE, 2002, pp. 350–352.
- [39] S. Asthana, N. Varma, V. K. Mittal, An investigation into classification of infant cries using modified signal processing methods, in: 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 2015, pp. 679–684. doi:10.1109/SPIN.2015.7095282.
- [40] M. S. Rusu, S. S. Diaconescu, G. Sardescu, E. Bratila, Database and system design for data collection of crying related to infant's needs and diseases, in: 2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), 2015, pp. 1–6. doi:10.1109/SPEd.2015.7343077.
- [41] C. E. Parsons, K. S. Young, M. G. Craske, A. L. Stein, M. L. Kringelbach, Introducing the oxford vocal (oxvoc) sounds database: a validated set of non-acted affective sounds from human infants, adults, and domestic animals, *Frontiers in psychology* 5 (2014) 562.
- [42] G. Zamzmi, C.-Y. Pai, D. Goldgof, R. Kasturi, Y. Sun, T. Ashmeade, Machine-based multimodal pain assessment tool for infants: A review, arXiv preprint arXiv:1607.00331.
- [43] A. Chittora, H. A. Patil, Data collection of infant cries for research and analysis, *Journal of Voice*.
- [44] Youtube, Statistics.  
URL <https://www.youtube.com/yt/press/statistics.html>
- [45] Youtube, Fair use guidelines (2016).  
URL <https://www.youtube.com/yt/copyright/fair-use.html>

- 
- [46] N. B. Ruparelia, *Cloud Computing*, Mit Press, 2016.
- [47] P. Mell, T. Grance, *The nist definition of cloud computing* (2009).
- [48] T. Burger, *The advantages of using virtualization technology in the enterprise*, 2012 (2013).
- [49] T. Christie, *Authentication*.  
URL <http://www.django-rest-framework.org/api-guide/authentication/>
- [50] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, *Http authentication: Basic and digest access authentication*, Tech. rep., RFC 2617 (1999).
- [51] T. Ganchev, N. Fakotakis, G. Kokkinakis, *Comparative evaluation of various mfcc implementations on the speaker verification task*, in: *Proceedings of the SPECOM*, Vol. 1, 2005, pp. 191–194.
- [52] M. Müller, *Information retrieval for music and motion*, Vol. 2, Springer, 2007.
- [53] J. O. Garcia, C. R. Garcia, *Mel-frequency cepstrum coefficients extraction from infant cry for classification of normal and pathological cry with feed-forward neural networks*, in: *Neural Networks, 2003. Proceedings of the International Joint Conference on*, Vol. 4, IEEE, 2003, pp. 3140–3145.
- [54] A. Zabidi, W. Mansor, L. Y. Khuan, R. Sahak, F. Y. A. Rahman, *Mel-frequency cepstrum coefficient analysis of infant cry with hypothyroidism*, in: *2009 5th International Colloquium on Signal Processing Its Applications*, 2009, pp. 204–208. doi:10.1109/CSPA.2009.5069217.
- [55] V. Tyagi, C. Wellekens, *On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition*, in: *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Spe-*

- ech, and Signal Processing, 2005., Vol. 1, 2005, pp. 529–532. doi: 10.1109/ICASSP.2005.1415167.
- [56] C. C. Aggarwal, A. Hinneburg, D. A. Keim, On the surprising behavior of distance metrics in high dimensional space, in: International Conference on Database Theory, Springer, 2001, pp. 420–434.
- [57] Wikipedia, Hiperravnina do podpornih vektorjev v metodi podpornih vektorjev (svm) za dva razreda.  
URL [https://en.wikipedia.org/wiki/Support\\_vector\\_machine#/media/File:Svm\\_max\\_sep\\_hyperplane\\_with\\_margin.png](https://en.wikipedia.org/wiki/Support_vector_machine#/media/File:Svm_max_sep_hyperplane_with_margin.png)
- [58] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE transactions on Neural Networks 13 (2) (2002) 415–425.
- [59] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Vol. 1, Springer series in statistics Springer, Berlin, 2001.
- [60] J. R. Quinlan, C4. 5: Programming for machine learning, Morgan Kaufmann (1993) 38.
- [61] W. D. H. T. . T. R. James, G., An Introduction to Statistical Learning, Springer, New York, 2013.
- [62] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
- [63] Skymind, A beginner’s guide to recurrent networks and lstms.  
URL <https://deeplearning4j.org/lstm>
- [64] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
- [65] S. V. Stehman, Selecting and interpreting measures of thematic classification accuracy, Remote sensing of Environment 62 (1) (1997) 77–89.

- 
- [66] T. Fawcett, An introduction to roc analysis, *Pattern recognition letters* 27 (8) (2006) 861–874.
- [67] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, et al., Orange: data mining toolbox in python, *Journal of Machine Learning Research* 14 (1) (2013) 2349–2353.
- [68] H. Wimmer, L. M. Powell, A comparison of open source tools for data science, *Journal of Information Systems Applied Research* 9 (2) (2016) 4.
- [69] J.-J. Aucouturier, Y. Nonaka, K. Katahira, K. Okanoya, Segmentation of expiratory and inspiratory sounds in baby cry audio recordings using hidden markov models, *The Journal of the Acoustical Society of America* 130 (5) (2011) 2969–2977.
- [70] S. Dieleman, B. Schrauwen, End-to-end learning for music audio, in: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 6964–6968.
- [71] K. Choi, G. Fazekas, M. Sandler, Automatic tagging using deep convolutional neural networks, *arXiv preprint arXiv:1606.00298*.
- [72] I. A. Bgnicg, H. Cucu, A. Buzo, D. Burileanu, C. Burileanu, Baby cry recognition in real-world conditions, in: *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, 2016, pp. 315–318. doi:10.1109/TSP.2016.7760887.
- [73] X. Zhou, D. Garcia-Romero, R. Duraiswami, C. Espy-Wilson, S. Shamma, Linear versus mel frequency cepstral coefficients for speaker recognition, in: *Automatic Speech Recognition and Understanding (ASRU)*, 2011 IEEE Workshop on, IEEE, 2011, pp. 559–564.
- [74] J. Sundberg, *The Science of the Singing Voice*, 1987.

- [75] Wikipedia, Acoustics (2016).  
URL [https://commons.wikimedia.org/wiki/File:Harmonic\\_partials\\_on\\_strings.svg#file](https://commons.wikimedia.org/wiki/File:Harmonic_partials_on_strings.svg#file)
- [76] M. Sawashima, H. Hirose, K. Honda, H. Yoshioka, S. Hibi, N. Kawase, M. Yamada, Stereoendoscopic measurement of the laryngeal structure, 1983, pp. 264–276.
- [77] C. Z. Boukydis, B. M. Lester, Infant crying: Theoretical and research perspectives, Springer Science & Business Media, 2012.
- [78] R. P. Daga, A. M. Panditrao, Acoustical analysis of pain cries' in neonates: Fundamental frequency, Int. J. Comput. Appl. Spec. Issue Electron. Inf. Commun. Eng ICEICE 3 (2011) 18–21.