

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019

Bc. Radoslav Kárník



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## ROZPOZNÁVÁNÍ HUDEBNÍCH NÁSTROJŮ ZE ZVUKOVÝCH NAHRÁVEK ZA POMOCI TECHNIK MUSIC INFORMATION RETRIEVAL

MUSICAL INSTRUMENTS RECOGNITION FROM AUDIO RECORDS USING MUSIC INFORMATION  
RETRIEVAL TECHNIQUES

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Radoslav Kárník

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Kiska

BRNO 2019



# Diplomová práce

magisterský navazující studijní obor **Audio inženýrství**

Ústav telekomunikací

**Student:** Bc. Radoslav Kárník

**ID:** 152699

**Ročník:** 2

**Akademický rok:** 2018/19

## NÁZEV TÉMATU:

### **Rozpoznávání hudebních nástrojů ze zvukových nahrávek za pomoci technik Music Information Retrieval**

#### POKyny PRO VYPRACOVÁNÍ:

V rámci této práce budou shrnuty dosavadní poznatky z oblasti zvané Music Information Retrieval (MIR). Cílem bude vytvořit databázi nahrávek hudebních nástrojů s co možno nejpočetnějším druhovým zastoupením. Dále bude navržen systém, který bude využívat metody MIR. Dále pak budou analyzovány nahrávky hudebních nástrojů z hlediska barvy zvuku, rytmiky a dynamiky. Rovněž budou vybrány parametry, pomocí kterých bude možné rozpoznat a určit jednotlivé hudební nástroje z audio nahrávek. K rozpoznávání hudebních nástrojů budou využity klasifikační algoritmy, pomocí kterých bude vyhodnocena diskriminační síla jednotlivých parametrů.

#### DOPORUČENÁ LITERATURA:

[1] ESSID, S., G. RICHARD a B. DAVID. Musical instrument recognition by pairwise classification strategies. Audio, Speech, and Language Processing, IEEE Transactions on [online]. USA: IEEE, 2006, 14(4), 1401-1412 [cit. 2018-09-07]. DOI: 10.1109/TSA.2005.860842. ISSN 1558-7916.

[2] MÜLLER, M. Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications [online]. Springer International Publishing Switzerland, 2015, 483 s. ISBN 978-3-319-21945-5.

**Termín zadání:** 1.2.2019

**Termín odevzdání:** 16.5.2019

**Vedoucí práce:** Ing. Tomáš Kiska

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Tato práce se zabývá návrhem a realizací klasifikačního systému pro rozpoznání hudebních nástrojů z hudebních nahrávek s využitím metod Music Information Retrieval. V první části práce jsou popsány jednotlivé parametry vhodné k rozpoznávání nástrojů, jejich výpočet z nahrávek a následná redukce příznakového vektoru. Další části jsou věnovány výběru, ladění a implementaci klasifikátorů v prostředí Python se zaměřením na neuronové sítě. Tyto klasifikátory jsou dále testovány na nahrávkách z databáze IRMAS, které obsahují 11 různých hudebních nástrojů hrající sólo anebo s dalšími nástroji. V poslední části jsou porovnány výsledky klasifikátorů při použití různých parametrů a různého počtu nástrojů.

## KLÍČOVÁ SLOVA

hudební nástroje, klasifikace, MIR, neuronové sítě, parametrizace, Python, strojové učení

## ABSTRACT

This paper discusses design and implementation of classifying system for recognition of musical instruments from audio records with use of Musical Information Retrieval techniques. In the first part, paper describes parameters used for instrument classification, calculation of said parameters from records and reduction of feature vector. Next part is devoted to tuning and implementation of various classifiers with focus on neural networks. These classifiers are further tested on records from IRMAS dataset which contain 11 musical instruments playing solo or with other instruments. Results of classifiers tested on different parameters and different numbers of instruments are discussed in the last part.

## KEYWORDS

classification, machine learning, MIR, musical instruments, neural networks, parametrization, Python

KÁRNÍK, Radoslav. *Rozpoznávání hudebních nástrojů ze zvukových nahrávek za pomoci technik Music Information Retrieval*. Brno, 2019, 72 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Tomáš Kiska

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Rozpoznávání hudebních nástrojů ze zvukových nahrávek za pomoci technik Music Information Retrieval“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Tomáši Kiskovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Purkynova 118, CZ-61200 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

podpis autora



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# Obsah

Úvod	12
<b>1 Hudební nástroje</b>	<b>13</b>
1.1 Hudební signál	13
1.1.1 Časové vlastnosti hudebního signálu	13
1.1.2 Frekvenční vlastnosti hudebního signálu	13
1.2 Dělení hudebních nástrojů	14
<b>2 Parametrizace hudebního signálu</b>	<b>17</b>
2.1 Music information retrieval	17
2.2 Vybrané parametry	17
2.2.1 Parametry popisující dynamiku	18
2.2.2 Parametry popisující barvu	19
2.3 Softwarové nástroje	24
2.3.1 MIRtoolbox	24
2.3.2 Sonic Visualiser	24
2.3.3 Sonic Annotator	25
<b>3 Metody klasifikace</b>	<b>26</b>
3.1 Klasifikátory	26
3.1.1 K-nejbližší soused	26
3.1.2 Gaussovy smíšené modely	27
3.1.3 Neuronové sítě	27
3.1.4 Náhodný les	29
3.1.5 Podpůrné vektory	30
3.2 Klasifikace nahrávek s více nástroji	31
3.2.1 Separace hudebních nástrojů	31
3.2.2 Multi-label klasifikátory	31
<b>4 Databáze nahrávek</b>	<b>33</b>
4.1 The University of Iowa MIS	33
4.2 IRMAS	33
<b>5 Výsledky zahraničních prací</b>	<b>35</b>
<b>6 Implementace</b>	<b>36</b>
6.1 Návrh systému	36
6.2 Příprava hudebních souborů	37



6.3	Předzpracování . . . . .	37
6.4	Parametrizace nahrávek . . . . .	38
6.5	Úpravy příznakového vektoru . . . . .	39
6.5.1	Normalizace dat . . . . .	40
6.5.2	mRMR . . . . .	40
6.5.3	Sequential forward floating search . . . . .	40
6.6	Klasifikace nahrávek s jedním nástrojem . . . . .	41
6.6.1	K-nejbližší soused . . . . .	41
6.6.2	Náhodný les . . . . .	41
6.6.3	Podpůrné vektory . . . . .	42
6.6.4	Neuronové sítě . . . . .	43
6.7	Klasifikace nahrávek s více nástroji . . . . .	47
<b>7</b>	<b>Výsledky práce</b>	<b>50</b>
7.1	Testování nahrávek obsahujících jeden nástroj . . . . .	50
7.1.1	Vliv počtu klasifikovaných tříd . . . . .	50
7.1.2	Vliv vybraných parametrů . . . . .	51
7.1.3	Testování dvojic nástrojů . . . . .	52
7.1.4	Výsledky klasifikace nahrávek obsahujících jeden nástroj . . . . .	53
7.2	Testování nahrávek obsahujících více nástrojů . . . . .	56
<b>8</b>	<b>Závěr</b>	<b>59</b>
	<b>Literatura</b>	<b>61</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>66</b>
	<b>Seznam příloh</b>	<b>67</b>
	<b>A Obsah přiloženého CD</b>	<b>68</b>
	<b>B Seznam parametrů získaných pomocí mRMR</b>	<b>69</b>
	<b>C Seznam parametrů získaných pomocí SFFS</b>	<b>71</b>

# Seznam obrázků

1.1	ADSR obálka hudebního tónu . . . . .	14
1.2	Spektrum piána. . . . .	15
2.1	RMS hudebního tónu . . . . .	18
2.2	Blokové schéma pro výpočet MFCC. . . . .	19
2.3	Příklad trojúhelníkových filtrů. . . . .	20
2.4	Doba náběhu $v$ na ADSR obálce. . . . .	22
2.5	Příklad zobrazení v aplikaci Sonic Visualiser. . . . .	24
3.1	Princip fungování kNN. . . . .	27
3.2	Popis neuronu . . . . .	28
3.3	Rozhodovací strom . . . . .	30
3.4	Podpůrné vektory . . . . .	30
6.1	Návrh systému. . . . .	36
6.2	Předzpracování nahrávky hudebního tónu. . . . .	38
6.3	Vliv počtu rozhodovacích stromů na úspěšnost klasifikace . . . . .	42
6.4	Vliv počtu rozhodovacích stromů na úspěšnost klasifikace . . . . .	42
6.5	Model NN použité při klasifikaci . . . . .	44
6.6	Průběh aktivační funkce ReLU . . . . .	44
6.7	Průběh aktivační funkce ReLU . . . . .	46
6.8	Model NN použité při ml klasifikaci . . . . .	47
6.9	Průběh aktivační funkce ReLU . . . . .	48
7.1	Vliv počtu tříd na úspěšnost klasifikace . . . . .	51
7.2	Vliv použitých parametrů na úspěšnost klasifikace . . . . .	53
7.3	Vliv počtu nástrojů v nahrávkách na $l_H$ a EMR . . . . .	58

# Seznam tabulek

1.1	Přehled hudebních nástrojů. . . . .	16
4.1	IRMAS - trénovací nahrávky. . . . .	34
4.2	IRMAS - testovací nahrávky. . . . .	34
6.1	Přehled použitých parametrů . . . . .	39
6.2	Příklad označení tříd . . . . .	39
6.3	Úspěšnost validace SVM . . . . .	43
7.1	Vliv počtu tříd na úspěšnost klasifikace . . . . .	51
7.2	Vliv použitých parametrů na úspěšnost klasifikace . . . . .	52
7.3	Výsledky klasifikace dvojic nástrojů s využitím NN . . . . .	54
7.4	Maticе záměn nástrojů. . . . .	55
7.5	Výsledky klasifikace nástrojů . . . . .	55
7.6	Ukázka multi-label predikce . . . . .	57
7.7	Vliv počtu nástrojů v nahrávkách na $l_H$ a EMR . . . . .	57
B.1	Seznam parametrů získaných pomocí mRMR 1 . . . . .	69
B.2	Seznam parametrů získaných pomocí mRMR 2 . . . . .	70
C.1	Seznam parametrů získaných pomocí SFFS 1 . . . . .	71
C.2	Seznam parametrů získaných pomocí SFFS 2 . . . . .	72

## Seznam výpisů

Ukazka_kodu_NN_sl.py . . . . .	46
Ukazka_kodu_NN_ml.py . . . . .	49

# Úvod

Tato diplomová práce se zabývá metodami klasifikace hudebních nástrojů z hudebních nahrávek za pomoci technik *Music Information Retrieval* neboli MIR.

Analýza hudebního obsahu má v dnešní době mnoho využití, a to hlavně při automatizaci různých úkonů, jako je například přepisování partitur, indexování a tagování hudebních médií v databázích pro doporučování obsahu, katalogích hudby, archivech a podobně. Tyto systémy pro svoji správnou funkčnost potřebují co největší množství informací. Extrakcí těchto informací z hudebních nahrávek se zabývá disciplína MIR.

Identifikace nástrojů může pomoci se zařazením hudební nahrávky do žánru či historického a geografického kontextu. Proto je potřeba vyvinout efektivní metodu zpracování hudebních signálů a extrakci relevantních parametrů, které charakterizují rozdílné hudební nástroje.

Práce se zaměřuje na analýzu poznatků ze zahraničních prací zabývajících se stejným problémem a následným výběrem osvědčených příznaků nesoucích relevantní informace o daném hudebním nástroji. Neméně důležitým faktorem v systému pro identifikaci hudebních nástrojů je použitý klasifikátor. V této oblasti jsou dnes populární neuronové sítě, které se dále rozšiřují díky dostupným počítačům s dostatečným výkonem. Jedna z kapitol se zabývá databázemi s dostatečným množstvím hudebních nahrávek k trénování a testování klasifikátoru.

V další části práce je popsán postup klasifikace od přípravy nahrávek, předzpracování, parametrizaci a výběru parametrů, až po výběr a vhodných klasifikátorů, jejich ladění, popis použitých hyperparametrů (to jsou parametry, které mění vlastnosti daných klasifikátorů) a v poslední řadě jejich implementaci v prostředí Python 3.6. Tato kapitola je dále rozdělena na popis klasifikace hudebních nahrávek, ve kterých se vyskytuje pouze jeden nástroj, a na tzv. multi-label klasifikaci, kdy testované nahrávky mohou obsahovat větší množství nástrojů.

Poslední část práce je věnována dosaženým výsledkům použitých klasifikátorů v případě single-label a multi-label klasifikace. Výsledky jsou zpracovány a dále porovnány na základě počtu nástroju v trénovacích a testovacích datasetech použitých při klasifikaci a na základě použitých skupin parametrů. V této části jsou také porovnány výsledky klasifikace dvojic hudebních nástrojů.

# 1 Hudební nástroje

Kapitola popisuje vznik hudebního signálu a jeho vlastnosti, které se dají charakterizovat pomocí příznaků popsanych v následující kapitole. Dále se věnuje dělení hudebních nástrojů do kategorií podle způsobu tvorby zvuku.

## 1.1 Hudební signál

Hudební nástroj je uměle vyrobený předmět vyrobený za účelem hudební reprodukce, který svou kmitavou energií vyzařuje do prostředí ve formě zvukové vlny. Vzniká tak hudební signál[22].

Hudební signál lze rozdělit na signál tónový a signál hlukový. U hlukového signálu převažuje šumová složka a je typický pro bicí nástroje. V nástroji zvuk vzniká pomocí těchto čtyř komponent[33]:

- **excitátor** — spouští proces tvorby tónu, dodává energii
- **oscilátor** — kmitá a tím vytváří zvuk
- **rezonátor** — zesiluje určité spektrální složky (formanty)
- **radiátor** — slouží k vyzařování zvuku z nástroje

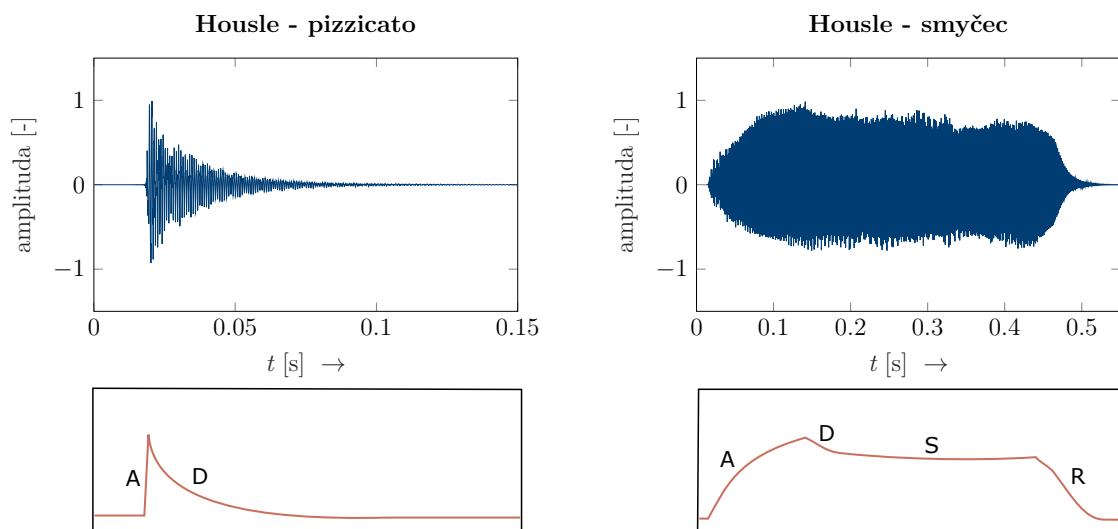
Typ excitátoru určuje hlavně charakter náběhu tónu. U perkusních nástrojů má buzení impulzní průběh (úder paličkou), u neperkusních nástrojů pozorujeme delší kontinuální průběh (saxofon). Rezonátor nefunguje jako klasický zesilovač, ale zdůrazňuje určité kmitočty ve spektru a spolu s radiátorem vyzařuje zvukovou energii do prostředí s minimálními ztrátami.

### 1.1.1 Časové vlastnosti hudebního signálu

Časový průběh signálu zobrazuje změny amplitudy závislé na čase. Časová obálka tohoto průběhu pak reprezentuje základní časové vlastnosti signálu a jeho délku. Základní části obálky jsou označeny jako ADSR podle jejich anglických názvů: A - attack (náběh), D - decay (útlum), S - sustain (pozdržení) a R - release (doznívání). Tato interpretace umožňuje rozlišit perkusní a neperkusní zvuky (obrázek 1.1). Z časového průběhu lze také odhadnout fundament (základní frekvenci) a energii tónu, tedy jeho hlasitost.

### 1.1.2 Frekvenční vlastnosti hudebního signálu

Hudební signál se skládá z fundamentu a vyšších harmonických, které jsou celočíselným násobkem fundamentu. Součástí je ale i šumová složka se spojitým spektrem s náhodným charakterem. Frekvenční spektrum tónu, tedy poměr a množství vyšších harmonických složek tónu a šumu ovlivňuje barvu zvuku, která je subjektivně



Obr. 1.1: ADSR obálka perkusního zvuku (drknutí na strunu houslí) a neperkusi-  
ního zvuku (tón byl na houslích zahrán smyčcem).

vnímána lidským uchem. Díky barvě zvuku jsme schopni rozeznat nástroje hrající  
stejný tón. Na obrázku 1.2 je zobrazeno spektrum tónu piána, ve kterém je možné  
vidět prvních osm harmonických složek (aliquót) [22].

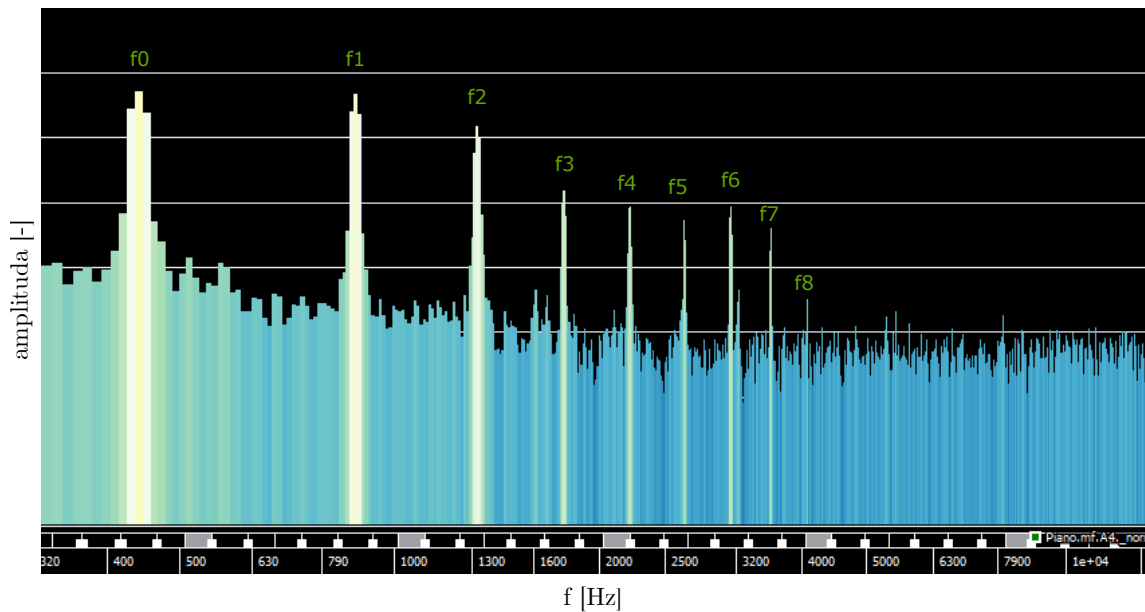
## 1.2 Dělení hudebních nástrojů

Hudební nástroje se historicky vyvíjely spolu se společností a kdysi populární druhy  
nemusí být v dnešní době používány. Jejich podoba, zvuk a složitost je také dána  
geograficky. V moderní společnosti se uchytilo několik nástrojů, které jsou součástí  
symfonických orchestrů a populárních hudebních skupin. Následující dělení se tedy  
kvůli přehlednosti zaměřuje pouze na těchto několik základních nejrozšířenějších  
nástrojů.

Nezákladnější dělení rozděluje nástroje na laděné (většina nástrojů, např. všechny  
strunné) a neladěné (bicí souprava), druhá skupina nástrojů není naladěna na urči-  
tou výšku tónu. Tato skupina tvoří hlukový (šumový) zvuk.

Dalším ze způsobů dělení hudebních nástrojů, který se v současnosti používá,  
je kategorizace podle vzniku tónů. Tedy podle toho, co svým kmitavým pohybem  
v nástroji vytváří zvukovou vlnu [22].

- **chordofony** — oscilátorem je struna napnutá mezi dvěma pevnými body
  - **smyčcové** — struna je uvedena do pohybu za pomoci smyčce
  - **drnkací** — struna je rozkmitána pomocí trsátka, nebo prstu
  - **kladívkové** — kmitání struny je způsobeno nárazem kladívka



Obr. 1.2: Spektrum piána hrajícího tón A4 o frekvenci 440 Hz. Toto grafické zobrazení je zachyceno v programu Sonic Visualiser (kap. 2.3.2).

- **membranofony** — oscilátor je tvořen membránou
- **aerofony** — oscilátorem je zvukový sloupec
  - **žestové** — k uvedení sloupce vzduchu do pohybu používají nátrubek
  - **plátkové** — vzduch je rozpořybován plátkem nebo dvojplátkem
  - **retné** — vzduch je rozpořybován nárazem na hranu nebo šřěrbinu
- **idiofony** — také nazývané jako samozvunčné - tyto nástroje vydávají zvuk vlastním chvěním
- **elektrofony** — jedná se o hudební nástroje, které využívají elektrickou energii
  - **elektrofonické** — kmitání mechanických oscilátorů je snímáno a elektrickou cestou dále upravováno a zesilováno
  - **elektronické** — nástroje, jejichž oscilátory jsou čistě elektronické

Tabulka 1.1 obsahuje základní zástupce ze všech výše zmíněných kategorií.



Tab. 1.1: Přehled základních zástupců kategorií hudebních nástrojů.

<b>Kategorie</b>	<b>Podkategorie</b>	<b>Zástupci</b>
chordofony	smyčcové	housle, viola, kontrabas
	drnkací a trsací	harfa, kytara, mandolína, banjo
	kladívkové a paličkové	cimbál, cembalo
membranofony		tympány, bubny, tamtam, tamburína
aerofony	žestové	trombón, pozoun, tuba, lesní roh
	plátkové	klarinet, saxofon, hoboj, fagot
	retné	flétna, pikola
idiofony		činely, vibrafon, zvonkohra, zvony, gong, tamtam, triangel
elektrofony	elektrofonické	elektrická kytara, hammondovy varhany, elektrické piano
	elektronické	syntetizéry, samplery

## 2 Parametrizace hudebního signálu

Tato kapitola se zabývá získáváním informací z hudebních digitálních nahrávek pomocí *Music information retrieval*. Zaměřuje se na příznaky, které jsou vhodné k parametrizaci nahrávek hudebních nástrojů pro jejich následnou identifikaci. V další části popisuje softwarové nástroje využívané k získávání těchto příznaků z digitálních nahrávek.

### 2.1 Music information retrieval

Zkráceně MIR, je vědní obor věnující se získání relevantní informace z hudebních digitálních médií. V podstatě se jedná o zpracování signálu v oblasti hudby. Získané informace se mohou použít na určování žánru, nástrojů, harmonie, tempa skladby atd.[27].

Tento obor se začal nejvíce rozvíjet v 90. letech 20. století hlavně díky digitalizaci hudby a přístupnosti dostatečně výkonných počítačů [4]. I když se MIR uplatňuje u různých médií, například obrazových dat (partitury) a metadat, tato práce se zaměřuje čistě na zvuková data v podobě digitálních nahrávek.

Při získávání hudebních informací z těchto audio souborů se nahrávka většinou nejdříve předzpracuje. Tento krok může zahrnovat normalizaci, sloučení vícekanálových nahrávek do jednoho kanálu, převzorkování a odstřížení částí nahrávek nenesoucí hudební informaci. Signál je pak rozdělen na překrývající se časové rámce o délce několika vzorků (záleží na vzorkovací frekvenci a příznaku, který se dále počítá). Tyto rámce jsou běžně vynásobeny okénkovou funkcí (nejčastěji Hammingovo anebo Hannovo okno), která kompenzuje vznik vysokofrekvenčních laloků ve spektru při dělení signálu.

### 2.2 Vybrané parametry

Tato podkapitola popisuje parametry, které se osvědčily v již uskutečněných pracích při identifikaci hudebních nástrojů. Vzhledem k tomu, že tempo a délka tónu jsou přímo dány tempem skladby, ve které nástroje zrovna hrají, nebereme parametry popisující tyto vlastnosti v potaz. Při rozpoznávání hudebních nástrojů se nejvíce osvědčily příznaky popisující barvu zvuku, a to hlavně MFCC, které jsou v současnosti nejpoužívanější [20, 2, 9, 1].

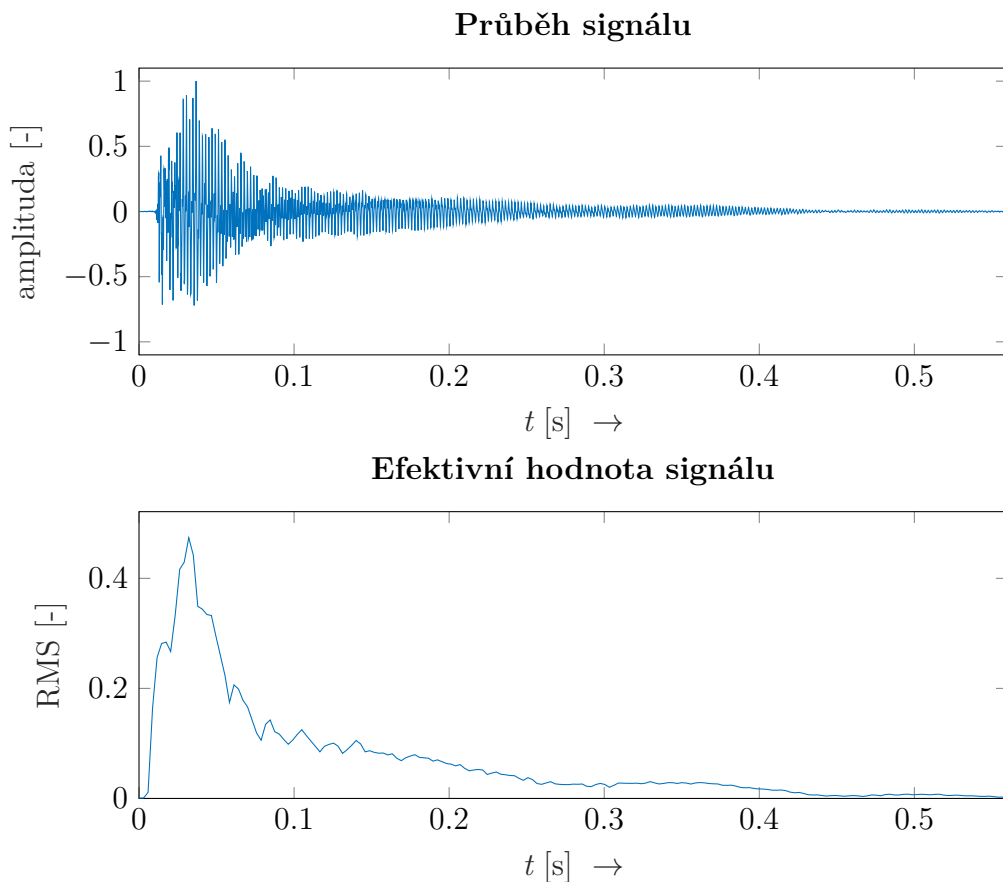
## 2.2.1 Parametry popisující dynamiku

### Efektivní hodnota signálu

*Efektivní hodnota* signálu  $RMS$  (*Root mean square*) udává energii a dynamiku signálu. Tento parametr se dále využívá například k výpočtu času náběhu (kap. 2.2.2). *Efektivní hodnota* jednoho časového rámce signálu o délce  $n$  je vypočítán, jako kvadratický průměr vzorků v tomto rámci [1]:

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}, \quad (2.1)$$

kde  $x_i$  je daný vzorek signálu.



Obr. 2.1: Průběh tónu violoncella a jeho efektivní hodnota.

### Nízká energie

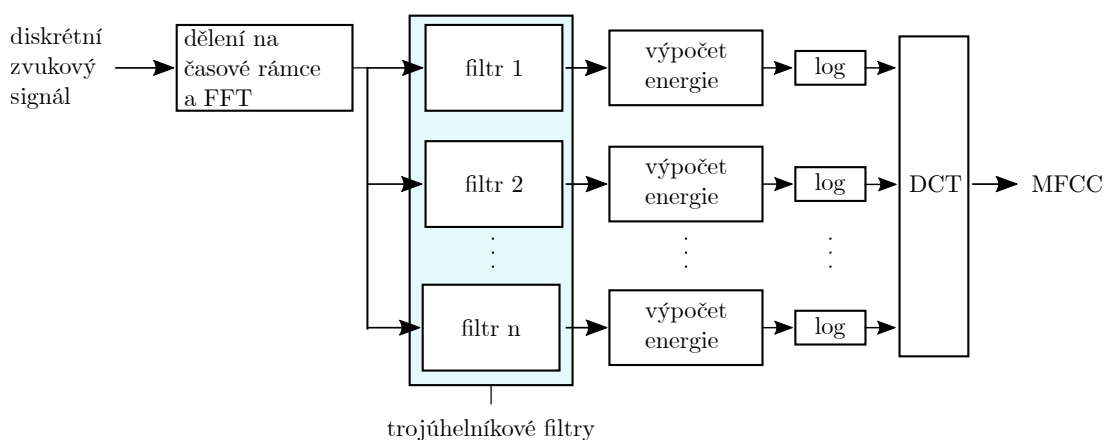
*Nízká energie* souvisí s efektivní hodnotou časových rámců signálu. Jedná se o procento časových rámců signálu, které mají energii (tedy výše uvedenou  $RMS$ ) menší než průměrnou[1].

## 2.2.2 Parametry popisující barvu

### Melovské frekvenční keprální koeficienty

*Melovské frekvenční keprální koeficienty* – *mel frequency cepstral coefficients*, dále *MFCC*, popisují krátkodobé spektrum signálu. Primárně používají v oblasti rozpoznávání řeči. V této oblasti se využívá jejich schopnost odhadnout tvar spektra a tedy i výskyt formantů.

*MFCC* ale mají dobré výsledky i v oblasti rozpoznávání hudebních nástrojů, kde se polohy formantů mohou výrazně lišit v závislosti na dynamice hry [9]. Rozložení spektrálních složek nám popisuje barvu zvuku nástroje, což je vlastnost, která je pro lidské ucho při rozeznávání nástrojů velmi důležitá.



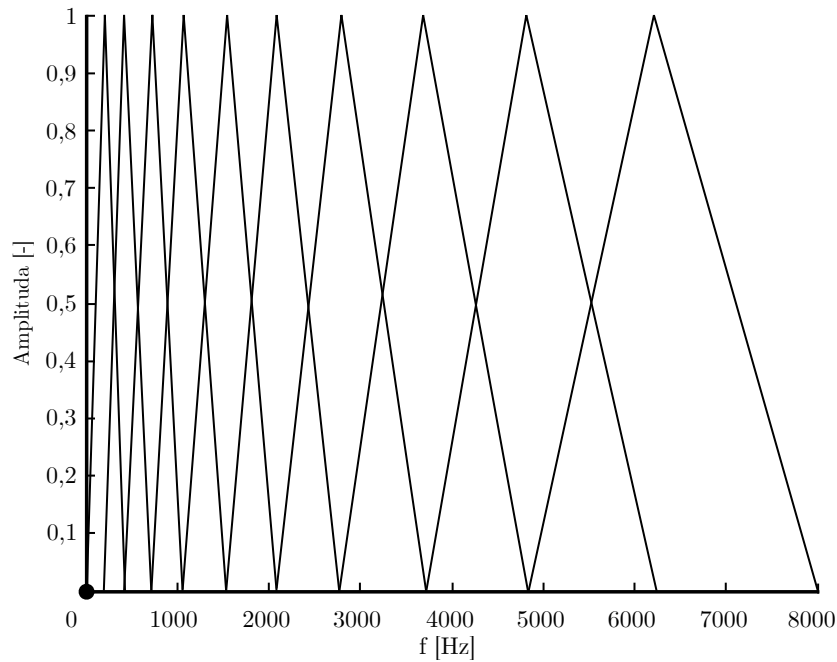
Obr. 2.2: Blokové schéma pro výpočet *MFCC* [31].

Každý segment rozděleného signálu (několik desítek milisekund) je nejprve převeden do frekvenční oblasti za pomoci FFT. Dále je vypočten modul spektra, který je filtrován bankou trojúhelníkových filtrů (obrázek 2.3). Ty jsou navrženy tak, aby výsledná škála odpovídala vnímání lidského ucha. Melovská škála se řídí podle následujícího vzorce [9]:

$$MEL(f) = 2595 \log\left(1 + \frac{f}{700}\right), \quad (2.2)$$

kde  $MEL(f)$  je frekvence v melovské škále a  $f$  [Hz] je frekvence vyjádřená lineárně. Pro každý filtr je vypočítán logaritmus energie a nakonec je melovské spektrum transformováno zpět do časové oblasti pomocí diskrétní kosinové transformace [16]. Výstupem jsou vektory čísel popisující jednotlivé segmenty hudebního signálu. Blokové schéma je zobrazeno na obrázku 2.2

Pro účely MIR se používají i parametry  $\Delta MFCC$  a  $\Delta\Delta MFCC$ , tedy první a druhá derivace *MFCC*. Ty udávají spektrální změny pro jednotlivé koeficienty



Obr. 2.3: Příklad deseti trojúhelníkových filtrů [25].

mezi jednotlivými časovými rámci [25]:

$$\Delta MFCC(t) = \frac{\sum_{n=1}^N n(c(t+n) - c(t-n))}{2 \sum_{n=1}^N n^2}, \quad (2.3)$$

kde  $c(t)$  je koeficient vypočítaný z časového rámce  $t$ ,  $N$  je číslo udávající, z jakého rámce relativně k rámci  $t$  se počítá derivace. V praxi se většinou používá  $N = 2$  nebo  $N = 3$ . Parametr  $\Delta\Delta MFCC$  se počítá obdobně s tím rozdílem, že se do vzorce místo koeficientů dosadí první derivace těchto koeficientů vypočítané z 2.3.

### Lineární predikční koeficienty

*Lineární predikční koeficienty LPC* jsou vypočítány pomocí lineárního predikčního kódování.

Aktuální vzorek signálu  $x(n)$  je vyjádřen jako lineární kombinace vzorků předcházejících a platí pro něj vztah [30]:

$$x(n) = \sum_{k=1}^p a_k x(n-k), \quad (2.4)$$

kde  $a_k$  je lineární predikční koeficient a  $p$  je řád *LPC*, tedy počet koeficientů a počet předcházejících vzorků použitých pro výpočet. Jednotlivé koeficienty nesou informaci o spektrálních vlastnostech zvukového signálu, proto se často používají pro rozpoznávání řeči.

## Konstantní Q transformace

Jedná se o metodu, která byla navržena pro výpočet frekvenčního spektra hudebních tónů. K výpočtu využívá logaritmicky škálované frekvenční rámce korespondující ke kmitočtům hudebních čtvrttónů. V jedné oktávě je tak 24 těchto rámců [18].

Narozdíl od FFT je každé časové okno jinak veliké v závislosti na frekvenčním rámci. Při použití *CQT* se nemění relativní mezery mezi harmonickými komponentami hudebních tónů. I když se tedy mění absolutní umístění frekvenčních složek při hraní různých tónů, jejich relativní pozice zůstává stejná, což je vhodná vlastnost pro rozpoznávání hudebních nástrojů.  $k$ -tá spektrální složka je dána následujícím vzorcem [18]:

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k, n] x[n] e^{-2\pi j \frac{Qn}{N[k]}}, \quad (2.5)$$

kde  $N[k]$  reprezentuje délku okna,  $W[k, n]$  Hammingovo okno,  $x[n]$   $n$ -tý vzorek signálu v daném okně a  $Q$  je činitel jakosti 1/24-oktávového filtru, rozlišení transformace je tedy jeden čtvrttón:

$$Q = \frac{1}{\sqrt[24]{2} - 1} \cong 34,14. \quad (2.6)$$

## Počet průchodů nulovou úrovní

*Zero crossing rate (ZCR)* je veličina, která nám udává, kolikrát zvukový signál projde nulou za určitý časový úsek — většinou za sekundu, nebo za zvolený časový rámec. Vypočítá se podle vzorce [7]:

$$ZCR = \frac{1}{2} \sum_{n=1}^{l-1} |\text{sign } x[n] - \text{sign } x[n-1]|, \quad (2.7)$$

kde  $x$  je vzorek signálu,  $l$  je počet vzorků ve vybraném časovém rámci a  $\text{sign } x$  je zjednodušená znaménková funkce:

$$\text{sign } x[n] = \begin{cases} +1 & \text{pro } x[n] \geq 0 \\ -1 & \text{pro } x[n] < 0 \end{cases}. \quad (2.8)$$

V MIR *ZCR* odlišuje zvuky s výraznou šumovou složkou (bicí) od zvuků čistších. V prvním případě *ZCR* nabývá vyšších hodnot [28].

## Čas náběhu

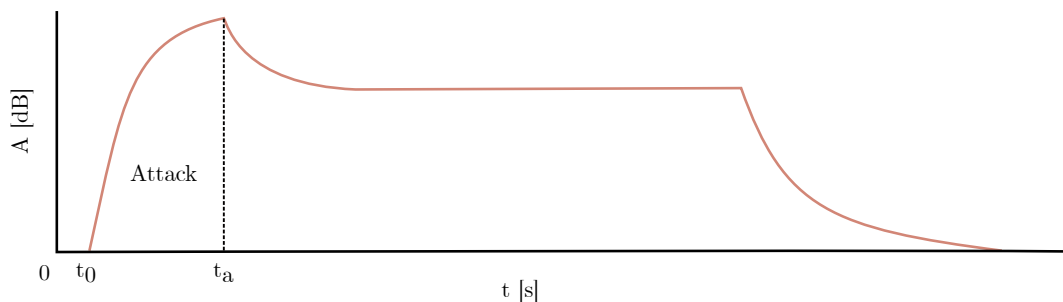
*Délka náběhu tónu* je doba, za kterou tón dosáhne nejvyšší intenzity. Na obrázku 2.4 je znázorněna ADSR křivka, která zobrazuje jednotlivé fáze hudebního tónu [31]. Podle Peeterse [28] se začátek a konec náběhu odhaduje pomocí času, kdy se energie obálky dostane na 20 % maximální energie pro začátek a 90 % maximální energie

pro konec náběhu. Kompenzuje tak možnou chybu vzniklou zašuměním signálu a situacemi, kdy se maximum obálky nevyskytuje na konci náběhu, ale později. To se vyskytuje například v obálce tónu zahraného na trubku.

V MIR se často používá log attack time, tedy desítkový logaritmus doby náběhu, vypočítá se podle vzorce

$$lat = \log_{10}(t_a - t_0), \quad (2.9)$$

kde  $t_a$  je čas dosažení 90 % maximální energie a  $t_0$  je čas na začátku hudebního tónu při dosažení 20 % energie [28].



Obr. 2.4: Zobrazení doby náběhu tónu na ADSR obálce.

### Spektrální centroid

*Spektrální centroid (spectral centroid)* je v podstatě střed spektra. Počítá se jako pravděpodobnostní rozdělení, jehož hodnoty jsou jednotlivé frekvence spektra a jejich pravděpodobnosti jsou jejich normalizované amplitudy ve spektru [28]:

$$\mu = \int x \cdot p(x) \delta x, \quad (2.10)$$

kde  $x$  jsou hodnoty jednotlivých frekvenčních složek:  $x = f(x)$  a  $p(x)$  je pravděpodobnost  $p(x) = \frac{a(x)}{\sum_x a(x)}$ .  $f(x)$  značí frekvenci a  $a(x)$  velikost dané spektrální složky.

### Spektrální rozložení

Tento parametr vyjadřuje *spektrální rozložení (spectral spread)* kolem spektrálního centroidu, vypočítá se jako rozptyl pravděpodobnostního rozdělení [28]:

$$\sigma^2 = \int (x - \mu)^2 \cdot p(x) \delta x, \quad (2.11)$$

přičemž  $x$  a  $p(x)$  jsou definovány výše v podkapitole 2.2.2.

## Spektrální šikmost

Šikmost spektra (*spectral skewness*) udává míru asymetrie vzhledem ke spektrálnímu centroidu. Vypočítá se ze třetího centrálního momentu  $m_3$ :

$$m_3 = \int (x - \mu)^3 \cdot p(x) \delta x, \quad (2.12)$$

šikmost pak dostaneme z následujícího vzorce [28]:

$$\gamma_1 = \frac{m_3}{\sigma^3}. \quad (2.13)$$

Hodnota šikmosti  $SK$  pak vyjadřuje tvar rozdělení spektra:

- $SKW = 1$  popisuje symetrické rozložení spektra
- pokud je  $SKW < 0$ , pak je větší množství energie spektra napravo
- pokud je  $SKW > 0$ , pak je větší množství energie spektra nalevo

## Spektrální špičatost

Spektrální špičatost (*spectral Kurtosis* —  $SK$ ) vyjadřuje míru špičatosti spektra okolo spektrálního centroidu a počítá se ze čtvrtého centrálního momentu [28]:

$$m_4 = \int (x - \mu)^4 \cdot p(x) \delta x, \quad (2.14)$$

špičatost je pak:

$$\gamma_2 = \frac{m_4}{\sigma^4}. \quad (2.15)$$

## Spektrální roll-off

Spektrální roll-off je frekvence, pod kterou se nachází určité procento energie signálu, většinou 85 %, nebo 95 %. Platí tedy vztah [28]:

$$\sum_0^{f_c} a^2(f) = 0,95 \sum_0^{f_s/2} a^2 \quad [\text{Hz}], \quad (2.16)$$

kde  $f_c$  je hledaná roll-off frekvence,  $a$  je velikost spektrálních složek a  $f_s$  je vzorkovací frekvence daného diskrétního signálu.

## Spektrální tok

Spektrální tok (*spectral flux*) měří velikost lokální změny ve spektru signálu. Je definován jako kvadrát změny velikostí odpovídajících si spektrálních složek mezi sousedními časovými rámci[2]:

$$SF = \sum_{k=2}^K (a(x) - a(x_{k-1}))^2, \quad (2.17)$$

kde  $a(x)$  je velikost  $x$ -té spektrální složky,  $k$  je pořadí časového rámce a  $K$  je počet časových rámců.



## 2.3 Softwarové nástroje

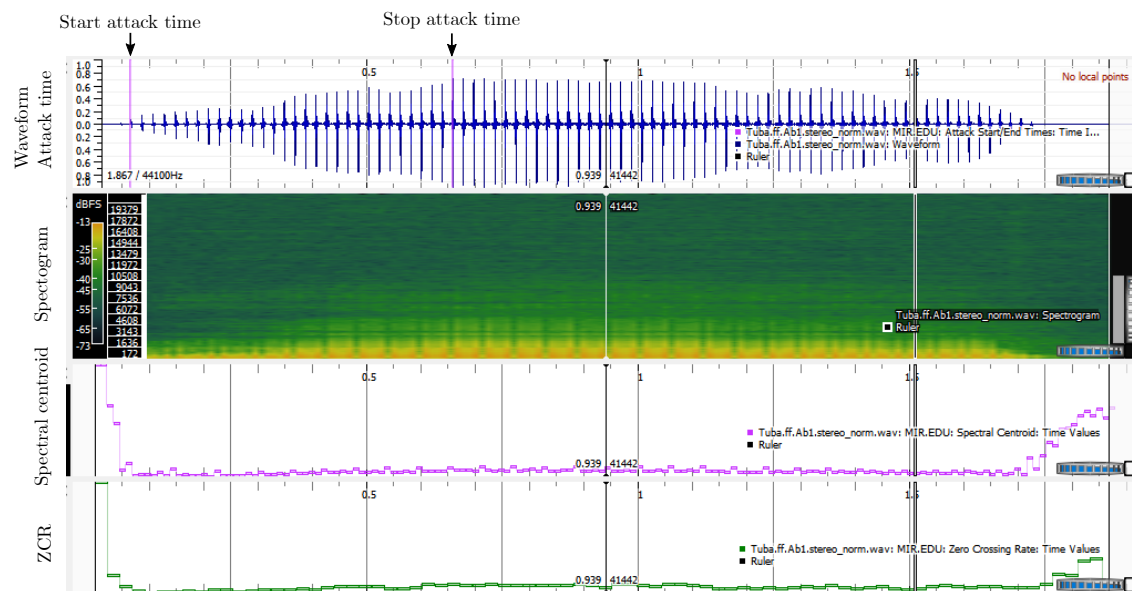
### 2.3.1 MIRtoolbox

MIRtoolbox je volně šiřitelný toolbox do prostředí Matlab, který obsahuje soubor funkcí sloužících k získávání dat ze zvukových nahrávek. Byl vyvinut jako součást projektu *Tuning the Brain for Music* v programu NEST (*New and Emerging Science and Technology*) [23]. Systém je navržen tak, aby spolu byly funkce navzájem kompatibilní a díky modularitě lze volat jak základní jednoduché funkce, tak i složitější funkce, které jsou využívány danými základními funkcemi.

Toolbox je z nahrávky schopen vyextrahovat až 50 různých audio příznaků patřících do časové i frekvenční oblasti a mimo jiné obsahuje také *Netlab toolbox* vhodný ke klasifikaci pomocí GMM (kap. 3.1.2) a *Auditory toolbox* používaný k výpočtu MFCC (kap. 2.2.2).

### 2.3.2 Sonic Visualiser

Jedná se o samostatný volně šiřitelný program pro analýzu a vizualizaci hudebních nahrávek vyvíjený v C++ [6]. Disponuje uživatelským rozhraním, které je podobné klasickým editačním programům, což mu umožňuje znázornit zvolené parametry audio nahrávky přímo v programu bez nutnosti přesunu dat do jiného prostředí.



Obr. 2.5: Sonic Visualiser — zobrazení parametrů na nahrávce tónu tuby.

Základní program umí zobrazit pouze tvar vlny signálu a spektrogram, jeho výhoda se ale skrývá v podpoře tzv. *Vamp plug-inů*, které se nachází na webové stránce

[www.vamp-plugins.org](http://www.vamp-plugins.org). Jedná se o plug-iny, které slouží k výpočtu a vizualizaci nespočtu MIR příznaků. Program také umožňuje vložení pluginů do vlastních panelů (panes), díky čemu se dají pozorovat souvislosti mezi jednotlivými příznaky.

### 2.3.3 Sonic Annotator

Sonic Annotator používá *Vamp plug-iny* stejně jako *Sonic Visualiser*. Jedná se ale o konzolovou aplikaci, jejíž vstupem jsou vždy tyto parametry:

- `-t` – název transformace, tedy název použitého plug-inu nainstalovaného na PC a nastavení transformace. Zde se zadává cesta ke konfiguračnímu souboru, který řídí parametry dané transformace. Při zadání příkazu `-d` místo `-t` je použito výchozí nastavení dané transformace.
- `-w` – výstupní formát výsledku – slouží k výběru formátu tabulky, do které jsou zapsány výsledky transformace
- cesta k audio souboru určenému k parametrizaci

Mezi výhody konzolové aplikace patří hlavně její nenáročnost a možnost ji volat z jiných programů, včetně skriptů psaných v *MATLABu* přes příkaz `system`.

## 3 Metody klasifikace

Všechny níže popsané klasifikátory využívají metody spadající do oboru *strojového učení* (*machine learning*). To je druh umělé inteligence, kdy se program snaží napodobit lidské myšlení a paměť. Hlavní vlastností strojového učení je schopnost učit se podle zkušeností nabytých z trénovacích vzorů a dále schopnost naučené vědomosti použít například při klasifikaci a regresi.

Učení se dělí na učení s učitelem a bez učitele. Při učení s učitelem je do systému zadán vstup a označený očekávaný výstup. Při učení bez učitele systém rozdělí vstupní informace na základě podobností a rozdílů aniž by tyto kategorie byly specifikované.

### 3.1 Klasifikátory

Klasifikátory jsou systémy, které rozhodují, do které skupiny (kategorie) patří nově sledovaná data. Tato část pojednává o často používaných klasifikačních systémech používaných v oblasti zpracování signálu.

#### 3.1.1 K-nejbližší soused

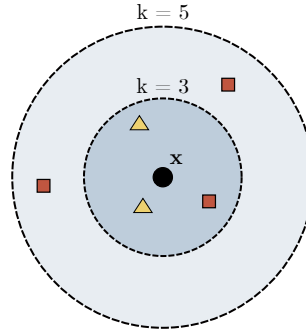
kNN, tedy *k-nejbližší soused*, je algoritmus, který počítá vzdálenost  $d(x, \boldsymbol{\nu}_i)$  mezi  $\boldsymbol{x}$  a každým příznakovým vektorem  $\boldsymbol{\nu}_i$ ,  $i \in \{1, 2, \dots, M\}$  z trénovacího setu, kde  $M$  je počet trénovacích vzorků. Pro výpočet vzdálenosti se používá *Eukleidovská vzdálenost* [20]:

$$d(x, \boldsymbol{\nu}_i) = \sqrt{\sum_{j=1}^D \left(x(j) - \boldsymbol{\nu}_i(j)\right)^2}, \quad (3.1)$$

kde  $D$  je rozměr příznakového vektoru. Po tom, co je  $d(x, \boldsymbol{\nu}_i)$  vypočítáno pro každý vektor  $\boldsymbol{\nu}_i$ , výsledky se seřadí od nejmenší po největší vzdálenost. Ve výsledku tedy  $k$  nejmenších hodnot odpovídá  $k$  nejbližším sousedům neznámého příznakového vektoru.

Jestliže  $k$  nejbližších sousedů obsahuje  $k_c$  příznakových vektorů náležící třídě  $c$ , potom daný vektor náleží třídě s největším  $k_c$ ,  $c \in \{1, 2, \dots, N\}$  a  $N$  je počet tříd.

Princip je zobrazen na obrázku 3.1, na kterém stejné útvary vždy odpovídají příznakovým vektorům náležící stejné třídě. Pokud zvolíme  $k = 3$ , vektor  $\boldsymbol{x}$  bude přiřazen třídě zeleného trojúhelníku, pokud ale  $k = 5$ , vektor  $\boldsymbol{x}$  bude přiřazen třídě červeného čtverce.



Obr. 3.1: Princip fungování kNN [1].

### 3.1.2 Gaussovy smíšené modely

GMM (anglicky *Gaussian Mixture Models*) využívají statistický způsob rozpoznávání na základě funkce rozložení hustoty pravděpodobnosti. Pro  $D$ -rozměrný příznakový vektor  $\mathbf{x}$  je hustota směsi použita pro pravděpodobnostní funkci definována jako [29]

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^M w_i p_i(\mathbf{x}). \quad (3.2)$$

Výsledný smíšený model rozložení hustoty pravděpodobnosti je váhovaná lineární kombinace  $M$  gaussových funkcí  $p_i(\mathbf{x})$ .

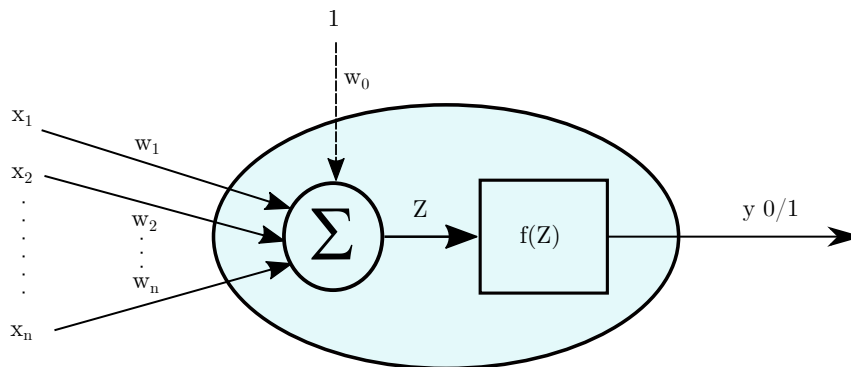
$$p_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)' \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right). \quad (3.3)$$

Parametry pravděpodobnostního modelu jsou označovány jako  $\lambda = \{w_i, \boldsymbol{\mu}_i, \Sigma_i\}$ , kde  $i = \{1, \dots, M\}$ ,  $w_i$  je váhovací parametr,  $\boldsymbol{\mu}_i$  vektor středních hodnot a  $\Sigma_i$  kovariační matice, přičemž platí, že  $w_i = 1$  a  $\Sigma_{i=1}^M$  [29].

### 3.1.3 Neuronové sítě

*Neuronová síť* je paralelní distribuovaný systém prvků modelujících biologické neurony. Ty jsou uspořádány tak, aby byl systém schopen požadovaného zpracování informací [17]. V dnešní době jsou neuronové sítě používány v široké škále úkonů využívajících klasifikaci a regresi – a to hlavně díky své modularitě.

Jednou z nevýhod neuronových sítí je poměrně složitý postup při vytváření modelu a volbě tzv. hyperparametrů, kdy se musí vhodně zvolit počet vrstev sítě, počet neuronů v každé vrstvě, rozhodovací funkce jednotlivých vrstev, ztrátová funkce, rychlost učení optimizera, způsob evaluace učení a v neposlední řadě počtem epoch, tedy cyklů učení. Při špatném zvolení těchto hyperparametrů může velmi lehce docházet k overfittingu (přeučení) sítě na trénovacím vzorku, nebo v opačném případě k underfittingu.



Obr. 3.2: Popis neuronu.

Na obrázku 3.2 je zobrazen princip fungování jednoho neuronu.  $y$  je binární výstup neuronu a je dán funkcí [17]

$$y = f \left( w_0 + \sum_{i=1}^n x_i w_i \right). \quad (3.4)$$

$f(z)$  je přenosová (aktivační) funkce neuronu, její výběr má vliv na výpočet výstupu neuronu. Mezi nejčastěji používané patří funkce lineární, skoková, Heavisideova, omezená, sigmoida a hyperbolická tangenta.  $z$  je vnitřní potenciál neuronu,  $w_0$  práh neuronu (bias),  $w$  jsou synaptické váhy neuronu,  $x$  jsou vstupy neuronu a  $n$  je počet vstupů.

## MLP

Pro potřeby klasifikace se často používají vícevrstvé perceptronové neuronové sítě (MLP, *multilayer perceptronal neural networks*). Jedná se o dopřednou neuronovou síť, ve které je soubor vstupů veden do odpovídajících výstupů pomocí vrstev, ty jsou minimálně tři: vstupní vrstva, skryté vrstvy a výstupní vrstva. Každý neuron nižší vrstvy je spojen se všemi neurony vrstvy vyšší. Každé spojení  $i$  je ohodnoceno synaptickou vahou  $w_{jk}$ , ta určuje propustnost signálu [2].

## Back propagation

Neuronové sítě se zpětnou propagací chyby jsou sítě, kdy jsou výsledky dopředné propagace šířeny zpětným průchodem a modifikují parametry přenosové funkce neuronu. Chyba sítě k  $i$ -tému tréninkovému vzoru je dána vztahem

$$E_i = \sum_{j=1}^c (d_{ij} - y_{ij})^2, \quad (3.5)$$

kde  $c$  je počet neuronů ve výstupní vrstvě,  $\mathbf{d}$  je žádaný výstup a  $\mathbf{y}$  je získaný výstup. Celková chyba sítě, tzv. míra naučenosti, se vypočítá jako suma chyb k jednotlivým tréninkovým vzorům  $r$  [17]:

$$E = \sum_{i=1}^r E_i. \quad (3.6)$$

Nová hodnota synaptických vah jdoucích z neuronu  $j$  do neuronu  $k$  ve výstupní vrstvě je

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t+1), \quad (3.7)$$

kde  $t$  je krok iterace a  $\Delta w_{jk}$  je změna váhy spojení:

$$\Delta w_{jk} = -\mu \sum_{i=1}^r \frac{\partial E_i}{\partial w_{jk}}, \quad (3.8)$$

kde  $\mu$  je zvolený koeficient učení,  $\frac{\partial E_i}{\partial w_{jk}}$  se spočítá pro každý tréninkový vzor  $p$ . Tato změna se vždy vypočítá pouze pro výstupní vrstvu sítě. Pro další vrstvy ji dostaneme zpětnou propagací [17].

### Proces učení MLP NN se zpětnou propagací

1. inicializace váhových hodnot  $w_0$  náhodnými malými čísly
2. předložení tréninkového vzoru a výpočet výstupu sítě
3. výpočet chyby sítě  $E_i$
4. testování na tréninkovou množinu
5. výpočet chyby ve výstupní vrstvě a zpětné šíření chyby
6. adaptace vah
7. ukončení procesu učení (pokud je chyba učení menší než zvolená  $E$ ), případně návrat ke kroku 2

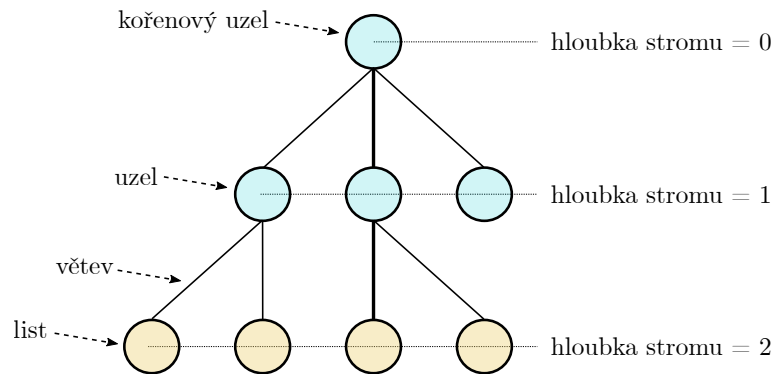
### 3.1.4 Náhodný les

Jedná se o učící algoritmus složený z několika rozhodovacích stromů, přičemž každý strom se učí z náhodně vybraných parametrů náhodně vybraných vzorků. Tím se snižuje korelace jednotlivých naučených stromů. Výsledek je potom dán sumarizací výsledků stromů. Tento algoritmus je často používán z důvodu jednoduchosti a možnosti použití při klasifikaci i regresi.

### Rozhodovací stromy

Tato metoda strojového učení je charakteristická svou hierarchickou strukturou (obr. 3.3). Uzel se na základě vstupního parametru a podmínky v uzlu rozdělí na několik dalších uzlů a z každého nového uzlu se tak stává nezávislý podstrom. Výhodou DT je flexibilita a jednoduchost implementace. Při vytváření DT do každého uzlu vstupují

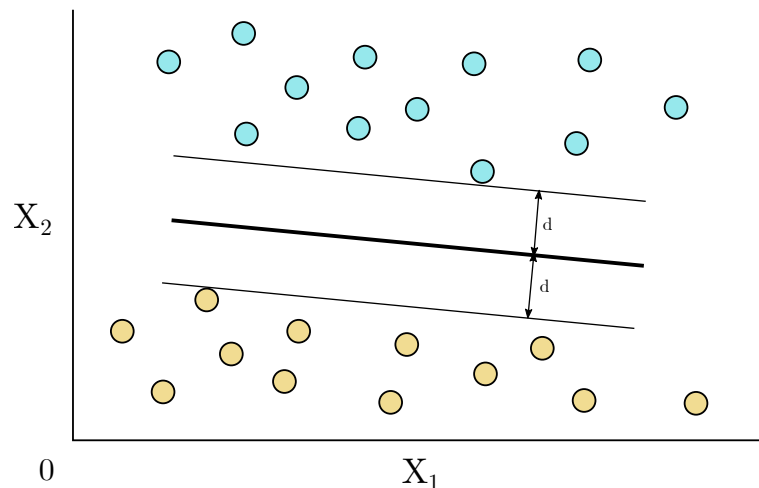
určitá data a pokud jsou splněny ukončovací podmínky (výsledek odpovídá třídě), je tento postup ukončen a z uzlu se stává list. V opačném případě se hledá způsob, jak stanovit podmínku, podle které budou data rozdělena do dalších uzlů [15].



Obr. 3.3: Struktura rozhodovacího stromu.

### 3.1.5 Podpůrné vektory

Metoda podpůrných vektorů (support vector machine – SVM) je metoda strojového učení s učitelem, která hledá nadrovinu rozdělující příznaky z trénovacích dat v prostoru (v podobě bodů odpovídajících jednotlivým příznakům) do dvou poloprostorů. Tato hledaná rovina  $d$ , představující co nejširší hranici mezi poloprostory, je dána souřadnicemi vektoru  $w$  a její poloha hodnotou prahu  $w_0$ .



Obr. 3.4: Příklad dělení prostoru pomocí algoritmu podpůrných vektorů

Jedná se tedy o binární klasifikátor, který rozlišuje pouze dvě třídy. Pro využití při klasifikaci více tříd se proto používají algoritmy rozdělující problém na jednotlivé

binární klasifikátory a to buď 1 vs 1 anebo 1 vs rest, kdy se porovnává každá dvojice tříd, případně jedna třída proti zbytku. Klasifikátor se pak rozhodne podle výstupů dílčích binárních klasifikátorů.

## 3.2 Klasifikace nahrávek s více nástroji

Výše popsané klasifikátory jsou primárně určeny pro klasifikaci dat, která jsou označena pouze jedním labelem a patří tedy pouze do jedné třídy. Pro případy nahrávek obsahujících více nástrojů tedy nestačí. V těchto případech máme možnost využít separaci signálů jednotlivých nástrojů z nahrávky, nebo využít klasifikátory určené pro tzv. multi-label klasifikaci.

### 3.2.1 Separace hudebních nástrojů

Pokud nastane případ, kdy testovaná nahrávka obsahuje více nástrojů, jedna z možností klasifikace nástrojů je separace jednotlivých nástrojů a následná klasifikace. Touto separací bez předem specifikovaných zdrojů se zabývá vědní disciplína nazývaná slepá separace signálů. I když neexistuje plně funkční metoda, níže zmiňované algoritmy vykazují dobré výsledky.

#### FASST

*Flexible Audio Source Separation Toolbox* je toolbox pracující v prostředí MATLAB určený k separaci zvukových zdrojů ze stereo zvukových nahrávek, který spojuje několik dříve používaných metod. Narozdíl od ostatních metod využívá jak rozdělení nástrojů ve stereu, tak i spektrální rozložení [26].

#### USCSS

Zkratka USSCS znamená *Unsupervised Single-Channel Separation of Nonstationary Signals*. Jedná se tedy o metodu, která separuje zdroje zvuku z jednobanových zvukových nahrávek. Stejně jako FASST, ani tento algoritmus nepotřebuje trénovací model. Systém je založen na časovo-frekvenční analýze a extrakci příznaků a na filtrování pomocí gammatone banky filtrů (gammatone filtr je lineární filtr popsáný impulsní odezvou, která vznikne součinem křivky gama rozdělení a sinusoidy) [12].

### 3.2.2 Multi-label klasifikátory

Další metodou je použití tzv. multi-label klasifikace, kdy jak trénovací, tak testovací data mohou být označeny pomocí labelu jedné a více tříd. Klasifikátor tak testovacím



datům nepřizuje pouze jednu třídu na úkor ostatních, ale rozhoduje o každé možné třídě zvlášť. V dnešní době se často používají neuronové sítě se speciální výstupní vrstvou a chybovým kritériem. V tomto případě je také nutné správné označení labelů tříd, které se narozdíl od single-label klasifikace zadává v podobě vektoru, jehož délka odpovídá počtu klasifikovaných tříd.

## 4 Databáze nahrávek

Pro učení a testování klasifikátorů je velmi důležité vybrat dataset s dostatečným množstvím dat pro dané klasifikované kategorie. Tato kapitola popisuje dvě často používané databáze s nahrávkami hudebních nástrojů, které jsou volně k dispozici.

### 4.1 The University of Iowa MIS

*The University of Iowa Musical Instrument Samples* [35] je databáze izolovaných samostatných tónů hudebních nástrojů, které byly nahrány v bezodrazové komoře. Tyto nahrávky jsou vytvářeny profesorem Lawrenceem Frittsem působícím na University of Iowa. Databáze je volně ke stažení již od roku 1997. Dataset obsahuje starší nahrávky nasnímané na mikrofon Neumann KM 84 a novější nahrávky, pro jejichž záznam byly použity dva mikrofony Earthworks QTC-40 napojené do decca tree. Všechny zvukové soubory mají vzorkovací frekvenci 44,1 kHz a bitovou hloubku 16 bitů. Ke každému nástroji patří několik zvukových souborů pokrývajících celý rozsah daného nástroje v různých intenzitách.

### 4.2 IRMAS

*Dataset for instrument recognition in musical audio signals* [3] je dataset složený z trénovacích a testovacích nahrávek deseti hudebních nástrojů a lidského zpěvu. Všechny nahrávky mají vzorkovací frekvenci 44,1 kHz a bitovou hloubku 16 bitů. Všechny tyto soubory jsou vystříhány z reálných hudebních nahrávek. Obsahují tedy velké množství šumu a zvuků, které mohou být při klasifikaci nástrojů způsobovat problémy a nepřesnosti. Na druhou stranu ale reprezentují skutečné využití hudebních nástrojů v populární hudbě. Proto je tato databáze vhodná pro praktické využití.

**Trénovací sada** obsahuje 6705 zvukových souborů o minimální délce tří sekund pocházejících z více než dvou tisíc nahrávek samostatných nástrojů (v názvu označeny značkou [nod]), případně nástrojů s bicími (označeny jako [dru]). Tabulka 4.1 uvádí počty testovacích nahrávek jednotlivých hudebních nástrojů, jejich značky uvedené v textových souborech sloužící k popsání obsahu nahrávek a index nástroje, který je dále používán v testovacích a trénovacích labelech jednotlivých nahrávek.

**Testovací sada** obsahuje 2891 nahrávek o délce mezi pěti a dvaceti sekundami. Nahrávky obsahují různé žánry s různou instrumentací. V testovací sadě se nevykytují žádné nahrávky ze sady trénovací. Každý soubor je označen podle žánru

Tab. 4.1: Přehled trénovacích nahrávek v databázi IRMAS.

hudební nástroj	značka	index	počet nahrávek
violoncello	[cel]	0	388
klarinet	[cla]	1	505
flétna	[flu]	2	451
akustická kytara	[gac]	3	637
elektrická kytara	[gel]	4	760
elektrické klávesy	[org]	5	682
piano	[pia]	6	721
saxofon	[sax]	7	626
trumpeta	[tru]	8	577
housle	[vio]	9	580
hlas	[voi]	10	778

Tab. 4.2: Přehled testovacích nahrávek v databázi IRMAS podle počtu nástrojů.

Počet nástrojů	1	2	3	4
Počet nahrávek	1236	1278	328	49

a nástrojů, které se v něm vyskytují: country-folk [cou-fo1], klasika [cla], pop-rock [pop-roc], latina-soul [lat-sou]. V každé nahrávce se vyskytují maximálně čtyři nástroje. Počty testovacích nahrávek rozdělené podle počtu nástrojů jsou zobrazeny v tabulce 4.2

## 5 Výsledky zahraničních prací

Otázkou identifikace hudebních nástrojů se za posledních několik let zabývalo poměrně velké množství týmů a jednotlivců. Většina k této problematice přistupovala podobně jako k identifikaci řečníků a proto také použili metody a příznaky, které se osvědčily v této oblasti.

Protože se od sebe nástroje nejvíce odlišují barvou zvuku, většina prací bere v potaz právě parametry popisující spektrální charakteristiku hudebního signálu. Dynamika a tempo je totiž závislé hlavně na typu skladby, ve které se nástroj vyskytuje a nepopisuje tak jeho obecné vlastnosti. Pro identifikaci nástroje jsou proto příznaky popisující tyto veličiny používány jen zřídka. K nejúspěšnějším příznakům patří MFCC, které používá většina prací, například Bhalke a Kazi [20, 2], Diment et al. [9] a další [1]. Dalšími příznaky, které vychází z identifikace a zpracování řečového signálu jsou lineární predikční koeficienty LPC [16]. Dalšími používanými příznaky jsou parametry popisující spektrum pomocí statistických metod. Mezi ně patří spektrální centroid, rozložení, šikmost, špičatost, roll-off, tok a další. Ve své práci je použili například Kaminskyj a Czasejk [18]. Ti k nim přidali CQT (konstantní Q transformace) a při klasifikaci metodou kNN dosáhli velmi dobrých výsledků. Mezi další parametry patří například čas náběhu a počet průchodů nulou ZCR [2].

Většina prací využívá jednu ze dvou databází popsanych v předešlé kapitole. Velmi důležitým faktorem ve výběru datasetu je počet nástrojů určených k identifikaci. Autoři prací [16, 1, 9] vybrali 8, respektive 9 nástrojů a úspěšnost jejich klasifikace při sestavení podobného příznakového vektoru byla vyšší než úspěšnost autorů, kteří klasifikovali až 23. nástrojů: [18, 20].

Úspěšnost identifikace je samozřejmě závislá i na výběru klasifikátoru. Ty nejpožívanější ve zmiňovaných pracích jsou popsány v kapitole 3. Z hlediska úspěšnosti nejlépe vychází systémy využívající neuronové sítě, které v závislosti na počtu parametrů často dosahují úspěšnosti nad 90 % [1]. Úspěšnost kNN se a GMM ale ve většině případů velmi nezaostává a pohybuje se zhruba od 60 % do 90 % [21, 20]. V pracích se ale vyskytuje mnoho proměnných ovlivňujících konečné výsledky a tak nelze jednoznačně určit, jaká část systému ovlivňuje jednotlivé výsledky.

Všechny již zmiňované zdroje ale nepočítají s možností, že je na testovaných nahrávkách přítomno více nástrojů, čímž se proces identifikace značně zkomplikuje. Tento problém řeší ve své dizertační práci Fuhrmann [10], který z vícenástrojové nahrávky nejdříve oddělil jednotlivé nástroje pomocí slepé separace (kapitola) a vzniklé nahrávky následně klasifikoval pomocí SVM (Support vector machine) s úspěšností kolem 50 %.

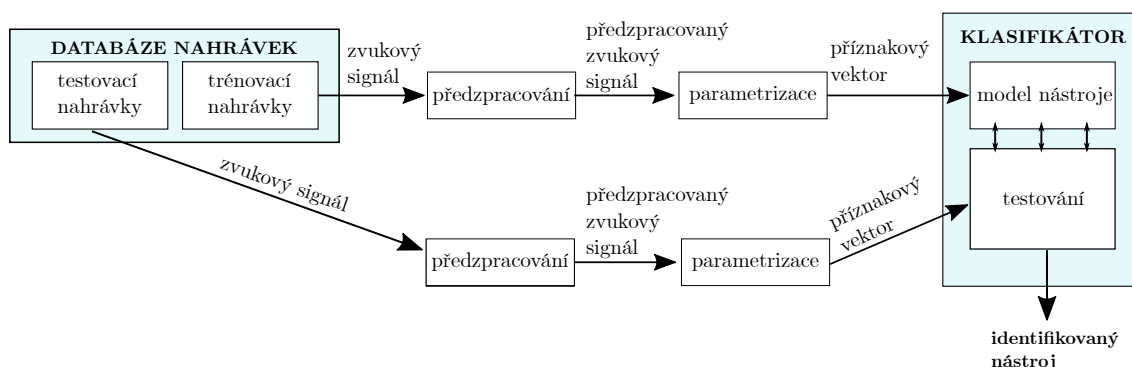
## 6 Implementace

Tato kapitola popisuje postup zpracování nahrávek a implementaci jednotlivých klasifikátorů na rozpoznávání hudebních nástrojů s použitím programů Matlab, Sonic Annotator a jazyku Python 3.6.

### 6.1 Návrh systému

Vlastní návrh systému na identifikaci hudebních nástrojů z hudebních nahrávek se opírá o poznatky z literatury v kapitole 5. Pro single-label klasifikaci byly vybrány klasifikátory využívající k-nejbližší sousedy, náhodné lesy, podpurné vektory a neuronové sítě. Pro multi-label klasifikaci byla použita neuronová síť s patřičným nastavením vhodným pro danou problematiku.

Na obrázku 6.1 je zobrazen návrh systému se všemi hlavními částmi, které jsou popsány v dřívějších kapitolách.



Obr. 6.1: Schéma systému pro identifikaci hudebních nástrojů.

Pro práci byla vybrána databáze *IRMAS* z důvodu velkého množství trénovacích a testovacích souborů. Narozdíl od databáze *The University of Iowa Musical Instrument Samples* také obsahuje úseky reálných skladeb místo jednotlivých not nástrojů nahraných v ideálních podmínkách bezodrazové místnosti. Proto je tato databáze vhodnější pro trénování a testování systému, který by měl klasifikovat nástroje v hudebních nahrávkách a to i v případě, kdy nástroj nehraje sám. S tím je však spojena očekávaná nižší úspěšnost klasifikace. Zvláště, když nahrávky obsahují bicí, hru na nástroje v různých dynamikách a mnoho rozdílných žánrů.

## 6.2 Příprava hudebních souborů

Jednotlivé nahrávky z trénovací i testovací složky byly pomocí programu `Matlab` nejprve hromadně přejmenovány podle odpovídajících textových souborů obsahujících tagy s názvy hudebních nástrojů, které byly součástí databáze. Tento krok zjednodušil budoucí práci s nahrávkami. Název zvukového souboru tak vždy obsahuje počet nástrojů následovaný jejich zkratkami, které jsou uvedeny v kapitole 4:

```
3_flu_gac_voi_04_Zeca_Baleiro_-_Com_Jards_Macale-2.wav
1_voi_radiohead_-_01_-_airbag-10.wav
4_flu_gel_pia_tru_01._Flight_Time-27.wav
2_sax_tru_01_-_Chet_Baker_-_Prayer_For_The_Newborn-24.wav
```

Testovací soubory také byly rozděleny podle počtu obsažených nástrojů, tedy od jednoho až po čtyři.

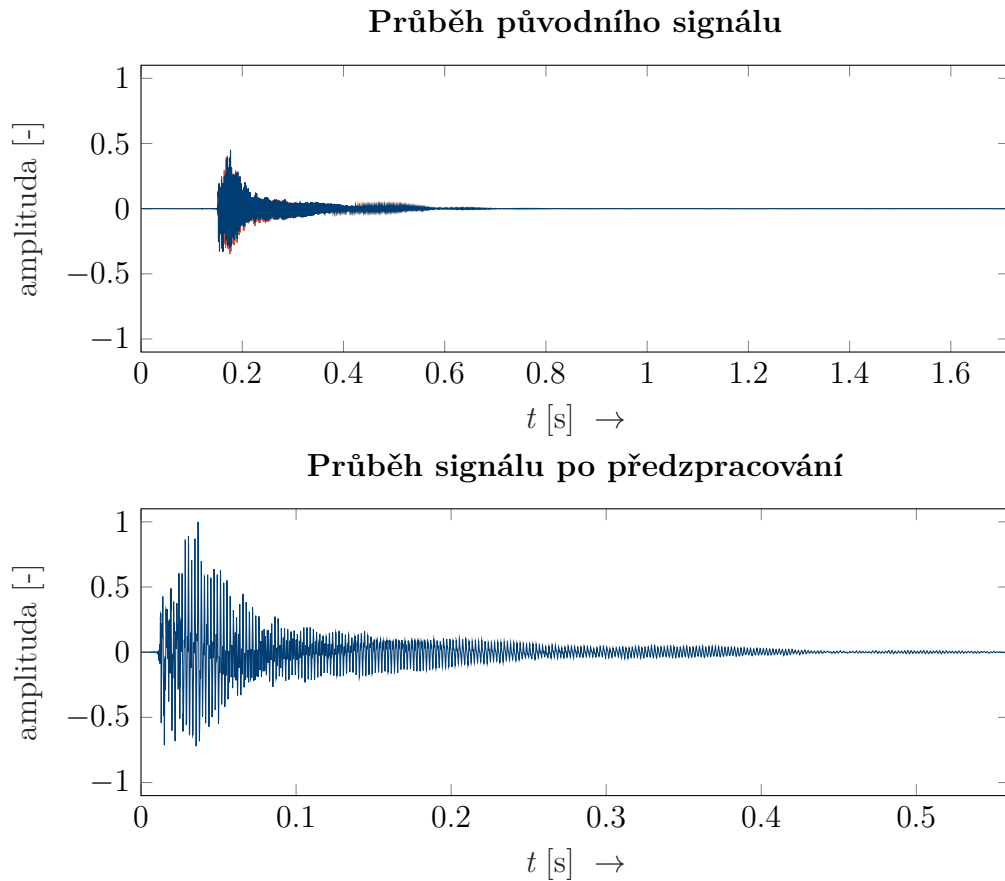
## 6.3 Předzpracování

Při předzpracování jsou všechny audio nahrávky hudebních tónů upraveny z důvodu redukce dat a optimalizace pro rychlejší výpočet. Podle povahy vstupních nahrávek mohou být provedeny následující operace:

- **Zkrácení** — V případě příliš dlouhé nahrávky se vzorek zkrátí na určenou délku několika sekund
- **Převedení na jeden kanál** — pokud je nahrávka vícekanálová, odpovídající vzorky kanálů se sečtou a vydělí počtem kanálů
- **Normalizace nahrávky** — každý vzorek nahrávky je vydělen maximem absolutní hodnoty signálu. Docílíme tím podobné hlasitosti všech nahrávek
- **Odstranění tichých míst** — signál je rozdělen na dostatečně dlouhé časové rámce, pokud je hodnota všech vzorků pod určitou úrovní (například 0,005), celý rámeček je z nahrávky odstraněn. Tím se zmenší velikost souboru a výpočet příznakových vektorů je tak méně náročný, aniž by došlo ke ztrátě informací
- **Převzorkování** — v případě různého anebo příliš vysokého vzorkovacího kmitočtu je možné jednotlivé nahrávky převzorkovat na zvolený kmitočet

Na obrázku 6.2 je zobrazeno porovnání signálu tónu zahraného pizzicatem na violoncello před předzpracováním a po něm. Na vrchním průběhu je vidět, že je signál dvoukanálový, obsahuje tichá místa a jeho nejhlasitější hodnoty nedosahují hodnoty 1. Na spodním průběhu je již signál předzpracovaný.

V této práci byly nahrávky normalizovány, převedeny na jeden kanál a byly z nich odstraněny případná tichá místa. Délka i vzorkovací kmitočet nahrávek byly ponechány.



Obr. 6.2: Porovnání nahrávky tónu violoncella před a po předzpracování.

## 6.4 Parametrizace nahrávek

V dalším kroku byly vypočítány vybrané parametry (kap. 2.2) ze všech nahrávek z datasetu s využitím konzolové aplikace `Sonic Annotator` volané z `Matlabu`. Při parametrizaci jsou parametry počítány ze všech časových rámců o zvolené délce s překryvem 50% vyfiltrovaných hammingovým oknem. Pro nahrávku rozdělenou na  $m$  rámců, pro které se počítá  $n$  parametrů, tak vznikla matice o rozměrech  $m \times n$ . Počítané parametry jsou uvedeny v tabulce 6.1. Primárně byly vybrány parametry popisující spektrum a barvu zvuku a vynechány ty, které popisují tempo a časovou oblast, protože se jedná o vlastnosti charakterizující spíše žánr hudby, než jednotlivé hudební nástroje.

Výsledný příznakový vektor každé nahrávky byl vypočítán pomocí statistických funkcí aplikovaných na odpovídající si parametry všech segmentů. Mezi tyto statistické funkce patří minimum, maximum, střední hodnota, směrodatná odchylka, median, první a třetí kvartil a první a devadesátý devátý percentil. Z výsledného vektoru byly odstraněny irelevantní hodnoty, jako je například minimum ZCR, jehož

Tab. 6.1: Přehled použitých parametrů.

Název parametru	Použitá zkratka	Počet koeficientů
konstantní Q transformace	cQ	48
lineární predikční koeficienty	LPC	13
melovské keprální koeficienty	MFCC	20
efektivní hodnota	RMS	1
nízká energie	LowE	1
spektrální centroid	s_centroid	1
spektrální špičatost	s_kurtosis	1
spektrální roll-off	s_rolloff	1
spektrální šikmost	s_skewness	1
spektrální rozložení	s_spread	1
počet průchodů nulovou úrovní	ZCR	1

hodnota se ve všech případech rovnala nule.

Takto vzniklé vektory byly pomocí `Matlabu` vloženy do pěti tabulek, kde první z nich byla vytvořená z trénovacích nahrávek a další čtyři z nahrávek obsažených v testovací databázi zvlášť podle počtu nástrojů. Pro budoucí analýzu diskriminační síly jednotlivých parametrů byly vždy zvlášť uloženy i vektory obsahující jednotlivé parametry. Před příznakový vektor byl vždy předřazen název nahrávky, číslo s počtem nástrojů a label, tedy štítek označující obsažené nástroje za pomocí tzv. one-hot kódování, kdy sloupce představují všechny možné třídy vyskytující se v databázi, nuly na řádku dané nahrávky vyjadřují, že se v dané nahrávce třída nevyskytuje, jedničky naopak označují její výskyt. Příklad je zobrazen v tabulce 6.2.

Tab. 6.2: Příklad označení tříd – nahrávka 3\_cel\_vio\_voi\_01\_Emily-10\_norm.wav

cel	cla	flu	gac	gel	org	pia	sax	tru	vio	voi
1	0	0	0	0	0	0	0	0	1	1

## 6.5 Úpravy příznakového vektoru

Velké množství extrahovaných parametrů může vést k příliš obsáhlému příznakovému vektoru, což vede k dlouhým výpočetním časům při klasifikaci. Následující



metody umožňují zmenšit rozměr vektoru s minimální ztrátou relevantních informací.

### 6.5.1 Normalizace dat

Normalizace dat, neboli standardizace, byla provedena před klasifikací pomocí funkce `StandardScaler` z knihovny `sklearn.preprocessing`. Tato funkce standardizuje každý parametr všech vektorů nezávisle na ostatních parametrech a vypočítá se odečtením střední hodnoty a následným vydělením směrodatnou odchylkou:

$$z = \frac{x - u}{s}, \quad (6.1)$$

kde  $x$  je normalizovaný vzorek,  $u$  je střední hodnota daného parametru (v naší matici sloupec) a  $s$  je směrodatná odchylka daného parametru.

### 6.5.2 mRMR

*Minimum-Redundancy-Maximum-Relevance* je algoritmus sloužící k filtraci parametrů z příznakového vektoru. Vybírá parametry, které nejlépe charakterizují danou třídu a jsou co nejrozdílnější od ostatních tříd. Vstupem tohoto algoritmu jsou příznakové vektory rozděleny do předem známých tříd a výstupem je vektor čísel vyjadřující relevantnost jednotlivých parametrů [13].

Data získaná z trénovacích souborů byla podrobena mRMR s omezením na 150 parametrů za účelem vyfiltrování dat, která pro budoucí klasifikaci nejsou relevantní. Seznam těchto 150 parametrů je uveden v příloze B.

### 6.5.3 Sequential forward floating search

Výběr parametrů pomocí *sekvenční dopředné selekce* (SFFS) je metoda používaná ve fázi předzpracování příznakového vektoru. Výběr se zaměřuje na set parametrů s co nejlepšími výsledky klasifikace, touto redukcí rozměru příznakového vektoru se také zvyšuje efektivita procesu klasifikace. Výběr je prováděn pomocí porovnávání výsledků klasifikace při přidávání a odebírání parametrů. Metoda hledá jednotlivé parametry, které jsou pro klasifikaci nejvhodnější, ale také se zaměřuje na jejich vhodné kombinace [11].

Jednou z nevýhod této metody je ale její vysoká výpočetní náročnost, která roste s počtem parametrů v příznakovém vektoru. Proto je v práci zařazena až za mRMR a vybírá tedy pouze ze 150 prvků. K implementaci této metody byl použita knihovna `mlxtend.feature_selection` využívající klasifikátoru náhodných lesů z knihovny `sklearn` v prostředí Python 3.6.

Po výstupu ze SFFS nám zbyl vektor o 100 parametrech. Jejich seznam je uveden v příloze C.

## 6.6 Klasifikace nahrávek s jedním nástrojem

Klasifikace nástrojů byla implementovaná pomocí programu PyCharm, který interpretuje jazyk Python 3.6. Ten nabízí množství knihoven vhodných ke strojovému učení. Jednou z nich je scikit-learn, zkráceně sklearn. Tato knihovna obsahuje nástroje pro klasifikaci, regresi, redukci rozměru a mnoho dalších. V této práci byl využit pro implementaci dříve popsané SFFS a klasifikaci pomocí kNN, RF a SVM.

Pro trénování a následné testování klasifikátorů pro nahrávky s jedním nástrojem byl formát labelů zjednodušen na jedno číslo od 0 do 10. Toto číslo vždy reprezentovalo jediný nástroj obsažený v nahrávce. Této problematice se říká *single-label* klasifikace. Vstupy klasifikátoru při trénování (funkce `fit`) tedy byly:

1. **vektor trénovacích labelů** o délce odpovídající počtu trénovacích nahrávek
2. **malice parametrů**, jejíž počet řádků odpovídal počtu trénovacích nahrávek a počet sloupců počtu použitých parametrů

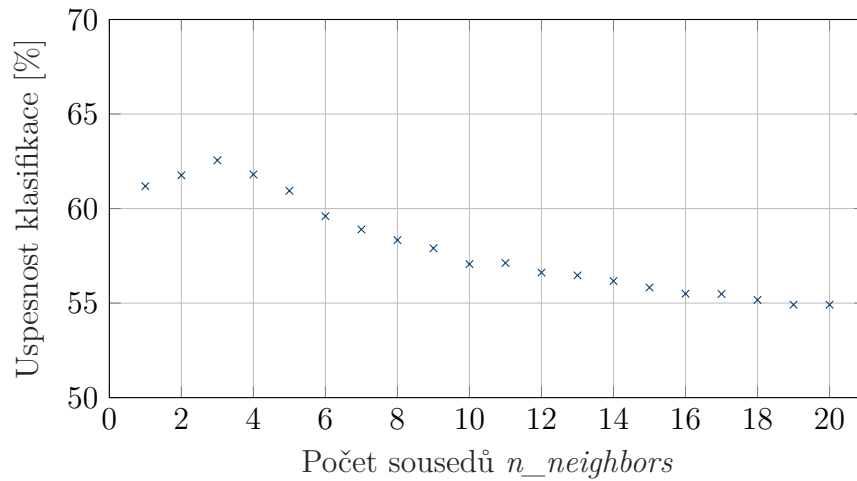
Vstupy funkce `predict` při testování klasifikátoru byly vždy obdobné s rozdílem použití testovacích nahrávek. Při validaci bylo využito metody grid search, kdy se mění tzv. hyperparametry (parametry klasifikátoru) a sleduje se úspěšnost v závislosti na těchto hyperparametrech. Všechny klasifikátory byly laděny při klasifikaci všech 11 nástrojů a jejich výstup je vždy vektor hodnot o délce rovnající se počtu tříd, přičemž za klasifikovaný nástroj je označena třída odpovídající pozici s největší hodnotou. K validaci klasifikátorů bylo vždy vybráno 20 % náhodných trénovacích nahrávek, které následně nebyly použity při učení.

### 6.6.1 K-nejbližší soused

Pro klasifikátor založený na k-nejbližších susedech byla použita třída `NearestNeighbors` z knihovny `sklearn`. Jediný laděný hyperparametr tohoto klasifikátoru je počet nejbližších prvků, které jsou brány v potaz při klasifikaci. Tento hyperparametr je značený jako `n_neighbors`. Ostatní nastavení bylo ponecháno ve výchozím nastavení. Nejlepších výsledků při validaci bylo dosaženo při `n_neighbors = 3` (viz graf 6.3).

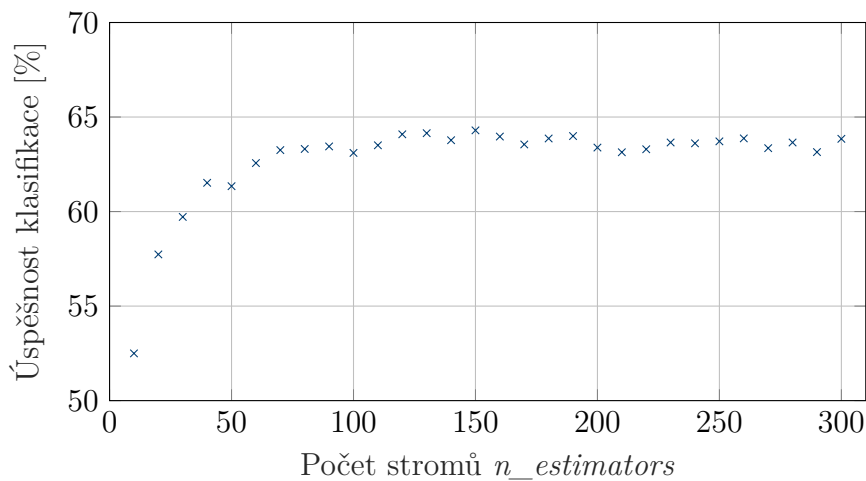
### 6.6.2 Náhodný les

Klasifikátor využívající náhodný les byl rovněž implementován pomocí `sklearn`. Ve třídě `sklearn.ensemble.RandomForestClassifier` byl opět měněn jediný hyper-



Obr. 6.3: Vliv počtu nejbližších sousedů na úspěšnost klasifikace validačních dat.

parametr  $n\_estimators$  značí počet rozhodovacích stromů v lese. Úspěšnost v závislosti na jejich počtu je vyobrazena v grafu 6.4, podle kterého bylo vybráno 150 stromů jako optimální množství pro klasifikaci.



Obr. 6.4: Vliv počtu rozhodovacích stromů na úspěšnost klasifikace validačních dat.

### 6.6.3 Podpůrné vektory

Stejná knihovna byla použita i pro implementaci podpůrných vektorů. Mezi nejdůležitější hyperparametry, které třída `sklearn.svm.SVC` umožňuje měnit, patří volba jádrové transformace, která se mění vstupním parametrem *kernel*.

Tab. 6.3: Úspěšnost klasifikace validačních dat v závislosti na volbě jádrové funkce.

kernel	linear	poly	rbf	sigmoid
ACC [%]	56,57	59,34	65,89	44,68

## 6.6.4 Neuronové sítě

Neuronové sítě byly implementovány za pomoci balíčku `TensorFlow`, přesněji jeho součástí `Keras` API. `Keras` umožňuje relativně jednoduché sestavování neuronových sítí bez nutnosti znalosti nízkourovňového programování a nabízí nástroje, které jsou uživatelsky přívětivé a dostačující pro skládání základních modelů neuronových sítí. Součástí je široký výběr hyperparametrů, jako jsou běžně používané optimizéry, aktivační funkce, chybová kritéria, regulizátory atd. knihovna také umožňuje pozorování průběhu učení neuronové sítě a nabízí možnost uchovávat si nejlepší nastavení a určovat kritéria, při kterých se má učení zastavit. Nesporonou výhodou při strojovém učení je i možnost paralelních výpočtů na grafické kartě díky knihovně `tensorflow-gpu`, která byla v práci využita.

### Vrstvy sítě

Volba nastavení modelu neuronové sítě probíhala taktéž pomocí evaluace úspěšnosti klasifikace validačních dat. Výsledný model je zobrazen na obrázku 6.5.

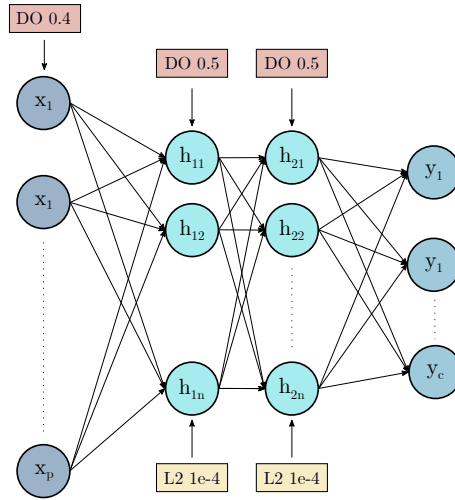
Počet neuronů  $p$  vstupní vrstvy  $x$  odpovídá délce příznakového vektoru, dále je model složen ze dvou skrytých vrstev  $h$  o  $n$  neuronech. Jako optimální počet neuronů ve skrytých vrstvách byla zvolena hodnota  $n \doteq 2/3p$ . Počet neuronů  $c$  ve výstupní vrstvě  $y$  odpovídá počtu klasifikovaných tříd:  $c = 11$ .

### Aktivační funkce a chybové kritérium

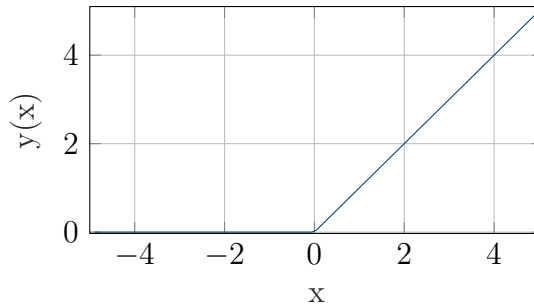
Pro neurony ve skrytých vrstvách byla použita aktivační funkce *ReLU* (*Rectified Linear Unit*). Jedná se o nejpoužívanější funkci počítanou podle vzorce

$$y(x) = \max(0, x). \quad (6.2)$$

Vstupní hodnoty menší než 0 jsou tedy na výstupu rovny nule, výstupní hodnoty kladných vstupních hodnot se nemění. Díky lineárnímu průběhu se jedná o jednoduchou aktivační funkci s rychlým výpočtem. Hlavní nevýhodou je ale náchylnost k úplné deaktivaci daného neuronu. V případě rychlého učení neuronové sítě při průchodu velké hodnoty může neuron své synaptické váhy aktualizovat skokově o velkou hodnotu a tím způsobit deaktivaci neuronu na další epoch [19]. Průběh funkce je zobrazen na grafu 6.6



Obr. 6.5: Model NN použitý při klasifikaci náhrávek s jedním nástrojem.



Obr. 6.6: Průběh aktivační funkce *ReLU*.

Ve výstupní vrstvě byla použita běžně používaná klasifikační funkce *softmax*, jejíž výstupy nabývají hodnot daných vzorcem [32]

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^c e^{x_j}} \quad (6.3)$$

Součet výstupů, jejichž počet je roven počtu klasifikovaných tříd  $c$ , je roven 1:

$$\sum_1^c y_i = 1 \quad (6.4)$$

Výstup funkce se tedy zaměřuje na maximalizaci hodnoty výstupu odpovídajícího právě jedné třídě. S aktivační funkcí *softmax* se pojí chybové kritérium *cross-entropy*, které bylo při trénování modelu použito. Hodnota chyby  $E_i$  pro tréninkový vzor  $x_i$  se pro jednotlivé výstupy vypočítá podle následujícího vzorce:

$$E_i = \sum_{j=1}^c d_j \log y_j - (1 - d_j) \log(1 - y_j), \quad (6.5)$$

kde  $d_j$  je tzv. target – tedy chtěná číselná hodnota odpovídající reálné třídě nástroje obsaženého v trénovací nahrávce [32]. Celková chyba sítě je vypočítána jako suma chyb k jednotlivým tréninkovým vzorům  $r$ :

$$E = \sum_{i=1}^r E_i. \quad (6.6)$$

## Optimizér

Jako nejvhodnější optimizér byl vybrán *Adam*. Jedná se o optimizér, který byl navržen speciálně pro hluboké neuronové sítě v roce 2014 a od té doby se stal jedním z nejpoužívanějších optimizérů. *Adam* ovlivňuje rychlost adaptivního učení tím, že hledá individuální rychlost učení pro každý parametr. K tomu využívá výpočet 1. a 2. momentu gradientu, jeho jméno tedy pochází z anglického *Adaptive moment estimation* [34]. Optimizér *Adam* byl využit v popisované neuronové síti s výchozí hodnotou učení o hodnotě  $10^{-3}$ .

## Prevence přeučení sítí

Z důvodu tzv. overfittingu (přeučení) sítě, kdy je úspěšnost predikce při učení mnohem větší než při validaci, byl ve všech vrstvách použit *dropout* (na obrázku 6.5 značený jako *DO*). Tato velmi efektivní metoda vždy vynechá určité procento náhodných neuronů ve vrstvě na základě pravděpodobnosti nastavené na dané vrstvě [19]. V tomto modelu byla použita pravděpodobnost 0,4 pro vstupní vrstvu a 0,5 pro skryté vrstvy. Ve výsledku se při trénování model učí z velkého množství menších modelů, kdy má každý z nich aktivní rozdílné neurony. Tím se snižuje jejich vzájemná závislost při rozhodování.

Další použitou metodou je *L2 regularizace*. Při aplikaci regularizace na vrstvu dojde k přičtení regularizačního prvku k chybové funkci. Upravená chybová funkce má pak tvar

$$E = E_0 + \frac{\alpha}{2} \sum_w w^2, \quad (6.7)$$

kde  $w$  jsou synaptické váhy,  $E_0$  je původní hodnota chyby bez regularizace a  $\alpha$  je parametr ovlivňující sílu regularizace [19]. V našem případě byla *L2* regularizace aplikována na obě skryté vrstvy přičemž byla zvolena hodnota  $\alpha = 10^{-4}$ .

## Implementace

Následující část kódu zachycuje implementaci takto nastaveného modelu sítě v prostředí *Python 3.6*. Model se první definuje pomocí přidávání vrstev se zvolenými počty neuronů, druhy aktivačních funkcí a regularizací. Následně se zkompiluje, v tomto kroku se deklaruje optimizér a chybové kritérium.

```

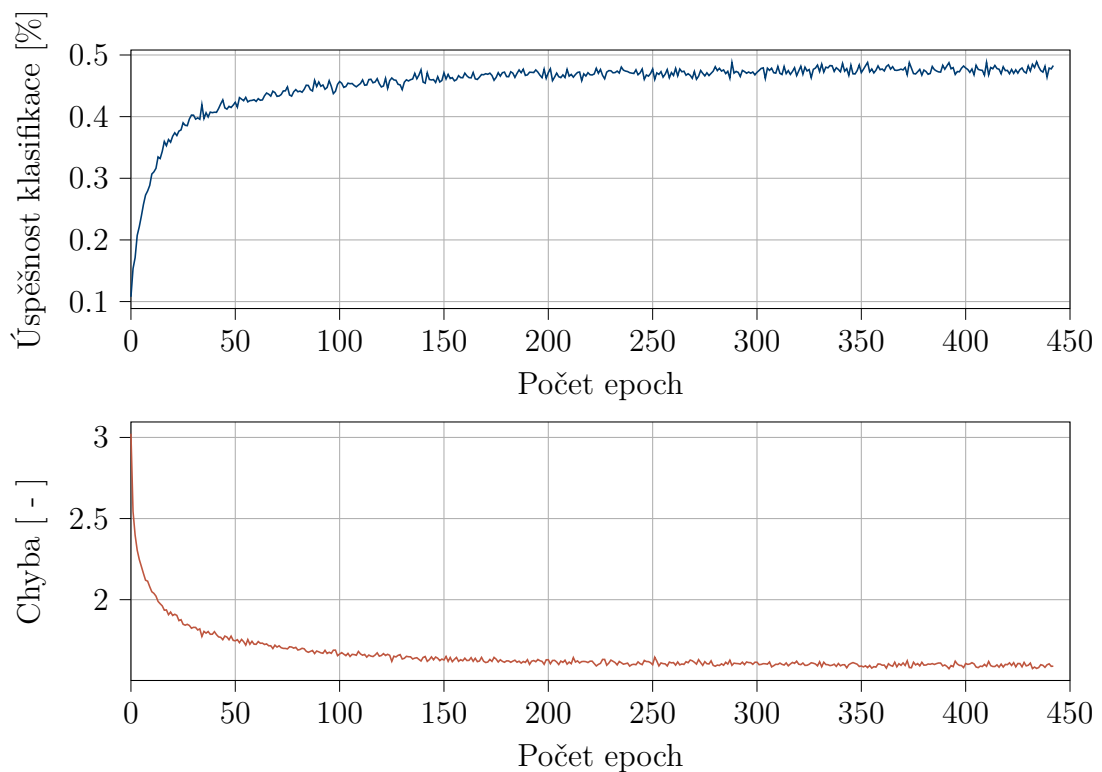
model = keras.Sequential()

model.add(keras.layers.Dropout(0.4))
model.add(keras.layers.Dense(n_neurons, input_dim=dimension, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_neurons, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_classes, activation='softmax'))

model.compile(optimizer=keras.optimizers.Adam(lr=0.001),
              loss='sparse_categorical_crossentropy', metrics=['accuracy'])

```

Při procesu učení byla v jednotlivých epochách sledována hodnota chyby a v případě vylepšení (snížení chyby) byly automaticky uloženy hodnoty synaptických váh neuronů. Díky tomuto opatření bylo uloženo nejlepší nastavení i v případě zhoršení klasifikační schopnosti sítě v následujících epochách. Pokud se hodnota nezlepšila po dobu 50 epoch, proces učení byl automaticky zastaven. Tato metoda je nazývána *early stopping*. Na obrázku 6.7 je zobrazen vývoj úspěšnosti klasifikace a chyby klasifikace validačních dat v závislosti na počtu prošlých epoch při trénování neuronové sítě.



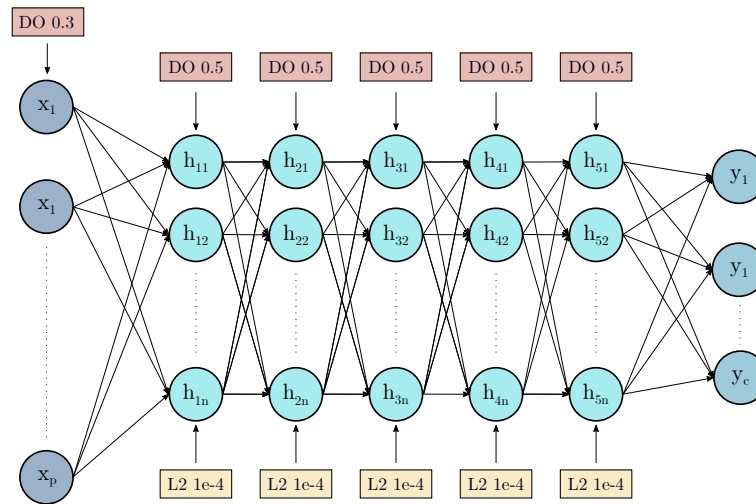
Obr. 6.7: Průběh aktivační funkce ReLU.

## 6.7 Klasifikace nahrávek s více nástroji

Pro klasifikaci nástrojů z nahrávek s více nástroji byla vybrána upravená neuronová síť z předchozí kapitoly a to díky možnostem ladění a nejlepším výsledkům při single label klasifikaci.

### Vrstvy sítě a optimizér

Do modelu byly při ladění přidány 3 skryté vrstvy s *ReLU* aktivační funkcí. Použitý model je zobrazen na obrázku 6.8. Počet neuronů  $p$  ve vstupní vrstvě  $x$  opět odpovídá délce příznakového vektoru, počet neuronů ve skrytých vrstvách  $h$  odpovídá  $n \doteq 2/3p$  a počet neuronů  $c$  ve výstupní vrstvě  $y$  se opět rovná počtu tříd, tedy 11. Nejlepších výsledků bylo opět dosaženo při použití optimizéru *Adam*.



Obr. 6.8: Model NN použitý při klasifikaci nahrávek s více nástroji.

### Aktivační funkce a chybové kritérium

Hlavním rozdílem je ale změna aktivační funkce výstupní vrstvy sítě. Při problematické multilabel klasifikaci se výsledky klasifikátoru nezaměřují pouze na jednu třídu, funkce *softmax* je tedy nepoužitelná. Místo ní se nejčastěji používá funkce *sigmoid* (*sigmoída*)

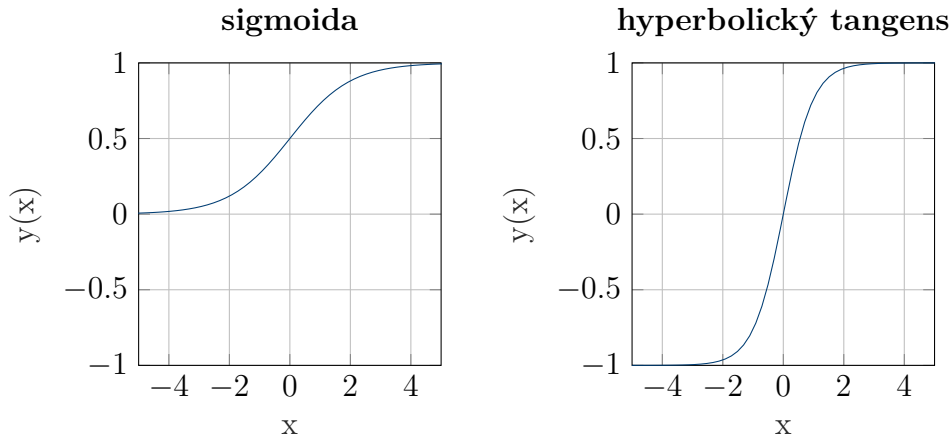
$$y(x) = \frac{1}{1 + e^{-x}}, \quad (6.8)$$

nebo *tanh* (*hyperbolický tangens*), prakticky se jedná o *sigmoidu* se změněným měřítkem na ose  $y$

$$y(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (6.9)$$



Tyto dvě funkce jsou zobrazeny na grafech an obrázku 6.9. V tomto modelu byla použita funkce  $\tanh$ , jejíž výstupní hodnoty nabývají hodnoty od -1 do 1, rozhodování při klasifikaci je tak pohodlnější. Výstupem této neuronové sítě je opět vektor hodnot o délce rovnající se počtu tříd. Tentokrát se ale za výsledek klasifikátoru nepovažuje třída odpovídající nejvyšší hodnotě, ale třídy odpovídající hodnotě, která je větší než 0. Chybové kritérium při multilabel klasifikaci musí brát ohled na všechny ob-



Obr. 6.9: Průběh aktivační funkce ReLU.

sažené a neobsažené třídy v trénovacím vzorku. I když je možné použít chybovou funkci popsanou ve vzorci 3.5, Zhang a Zhou přišli ve článku [34] se speciálním výpočtem chyby, který se nezaměřuje pouze na rozlišování jednotlivých labelů (jestli label  $j \in Y$  patří do nahrávky  $x_i$ ), ale bere v potaz i korelace mezi různými labely  $x_i$ . Třídy obsažené v  $Y_i$  jsou hodnoceny s větší vahou, než třídy neobsažené. Chybová funkce k  $i$ -tému tréninkovému vzoru je dána vztahem

$$E_i = \frac{1}{|Y_i| |\bar{Y}_i|} \sum_{(k,l) \in Y_i \times \bar{Y}_i} e^{-(c_{ik} - c_{il})} \quad (6.10)$$

kde  $\bar{Y}_i$  je doplněk labelů  $Y_i$   $i$ -tého tréninkového vzoru,  $(c_{ik} - c_{il})$  představuje rozdíl výstupu sítě jednoho labelu, který patří k  $x_i$  ( $k \in Y_i$ ) a jednoho, který do něj nepatří: ( $l \in \bar{Y}_i$ ). Celková chyba sítě se pak opět vypočítá jako suma chyb k jednotlivým tréninkovým vzorům  $r$ :

$$E = \sum_{i=1}^r E_i. \quad (6.11)$$

### Prevence přeučení sítě

Regularizace učení probíhala stejným způsobem, jako v případě klasifikace nahrávek s jedním nástrojem. Jediným rozdílem je změna pravděpodobnosti *drop outu* na

vstupní vrstvě z 0,4 na 0,3. Na všech vrstvách je opět použita *L2* regularizace s hodnotou parametru  $\alpha = 10^{-4}$ .

## Implementace

Při učení sítě se opět využilo *early stoppingu* a ukládání hodnot synaptických vah při nejlepších hodnotách chyby stejným způsobem, jako v předchozím modelu. Následující část kódu zobrazuje implementaci modelu sítě pomocí API Keras.

```
model = keras.Sequential()

model.add(keras.layers.Dropout(0.3))
model.add(keras.layers.Dense(n_neurons, input_dim=dimension, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_neurons, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_neurons, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_neurons, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_neurons, activation='relu',
                             kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(n_classes, activation='tanh'))

model.compile(loss=bp_mll_loss, optimizer=keras.optimizers.Adam(lr=0.001))
```

## 7 Výsledky práce

Tato kapitola popisuje výsledky, kterých bylo dosaženo použitím klasifikačních systémů popsaných v předešlé kapitole. Veškeré testování bylo prováděno na testovacím setu nahrávek, který nebyl použit při trénování, ani valdaci klasifikátorů.

### 7.1 Testování nahrávek obsahujících jeden nástroj

#### 7.1.1 Vliv počtu klasifikovaných tříd

V této části byla tedy sledována změna úspěšnosti klasifikace čtyř klasifikátorů při přidávání (respektive odebrání) nahrávek s jednotlivými nástroji do trénovacího a testovacího datasetu. Použity byly všechny dostupné parametry vyfiltrované metodami mRMR a SFFS. K jednoduché evaluaci klasifikátorů je použita celková úspěšnost, neboli přesnost:

$$\text{Úspěšnost} = \frac{\text{počet správně hodnocených tříd}}{\text{počet testovaných nahrávek}} \cdot 100 \quad [\%] \quad (7.1)$$

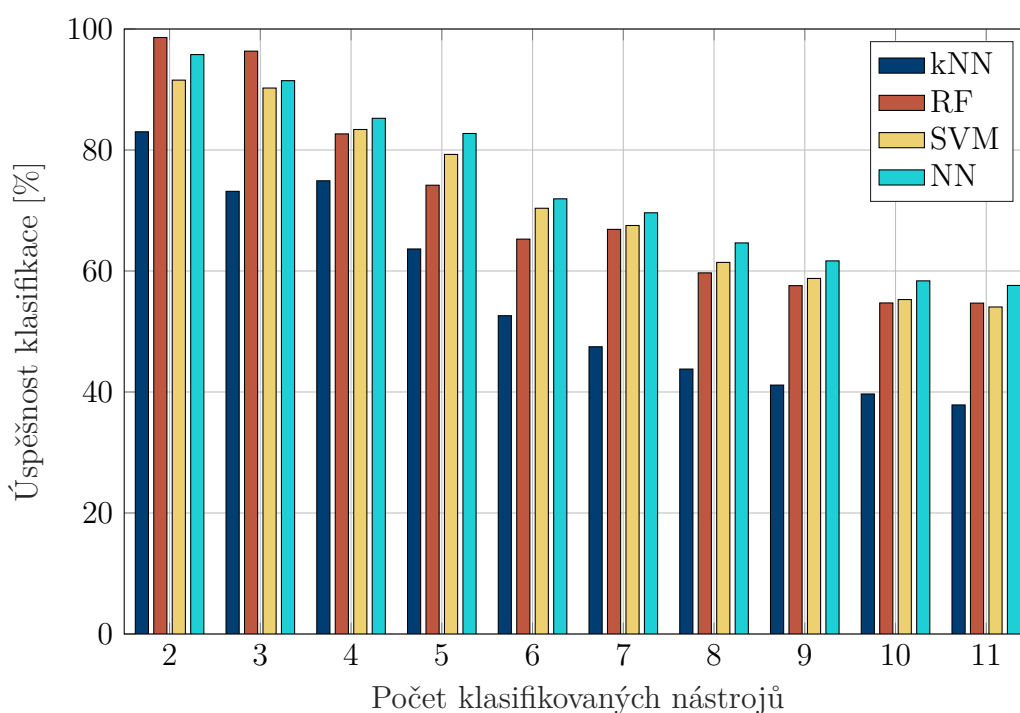
Nástroje byly přidávány postupně podle úspěšnosti klasifikace od nejlepších po nejhorší: jako první byla testována dvojice housle – violoncello a dále byly přidávány nástroje v následujícím pořadí:

1. trumpeta
2. akustická kytara
3. zpěv
4. elektrická kytara
5. piano
6. varhany
7. klarinet
8. flétna
9. saxofon

Výsledky jsou uvedeny v tabulce 7.1 a zobrazeny ve sloupcovém grafu 7.1. Výsledné hodnoty potvrzují hypotézu, že úspěšnost klasifikace je nepřímo úměrná počtu nástrojů, kdy se zvyšuje počet možností, mezi kterými se klasifikátor rozhoduje. Dlouhodobě dobré výsledky vykazuje neuronová síť, kterou překonal pouze klasifikátor s náhodným lesem při menším počtu nástrojů (2 a 3). Obecně je ale druhým nejlepším klasifikátorem systém využívající podpurné vektory. Nejnížší úspěšnost s poměrně velkým rozdílem ve všech situacích vykazuje klasifikátor kNN.

Tab. 7.1: Vliv počtu tříd na úspěšnost různých klasifikátorů.

počet nástrojů	klasifikované nástroje	úspěšnost klasifikace [%]			
		kNN	RF	SVM	NN
2	housle, violoncello	83,01	<b>98,59</b>	91.55	95.77
3	+ trumpeta	73,17	<b>96,34</b>	90.24	91.46
4	+ akustická kytara	74,91	82.66	83.39	<b>85.24</b>
5	+ zpěv	63,64	74.18	79.27	<b>82.73</b>
6	+ elektrická kytara	52,61	65.27	70.37	<b>71.93</b>
7	+ piano	47,48	66.88	67.52	<b>69.62</b>
8	+ varhany	43,79	59.69	61.42	<b>64.64</b>
9	+ klarinet	41,14	57.58	58.77	<b>61.67</b>
10	+ flétna	39,67	54.72	55.28	<b>58.37</b>
11	+ saxofon	37,86	54.69	54.05	<b>57.61</b>



Obr. 7.1: Vliv počtu tříd na úspěšnost různých klasifikátorů.

## 7.1.2 Vliv vybraných parametrů

V další části probíhal test úspěšnosti klasifikátorů v závislosti na vybraných parametrech. Ty byly rozděleny do pěti samostatných skupin uvedených v tabulce 7.2.

Do skupiny *spektrální parametry* patří spektrální centroid, spektrální šikmost, spektrální špičatost, spektrální roll-off a spektrální tok. Při testování všech parametrů bylo opět využito metod mRMR a SFFS a všechny tyto testy byly prováděny při klasifikaci všech 11 nástrojů.

Výsledky ukazují, že klasifikátory vykazují rozdílné úspěšnosti při použití jednotlivých skupin parametrů. To je nejvíce zřejmé na první skupině obsahující RMS a nízkou energii. Neuronová síť si v porovnání s ostatními systémy s touto skupinou poradila velmi dobře. Výsledky ostatních klasifikátorů jsou horší a to s poměrně velkými rozdíly. V případech ostatních skupin parametrů se úspěšnost liší mnohem méně (až na kNN). Ve většině případů je nejúspěšnější neuronová síť, velmi dobře si ale vede i klasifikátor RF, který má ve dvou případech dokonce největší úspěšnost klasifikace. Výsledky podpurných vektorů jsou velmi podobné, až na již zmiňovanou první skupinu obsahující parametry RMS a nízkou energii.

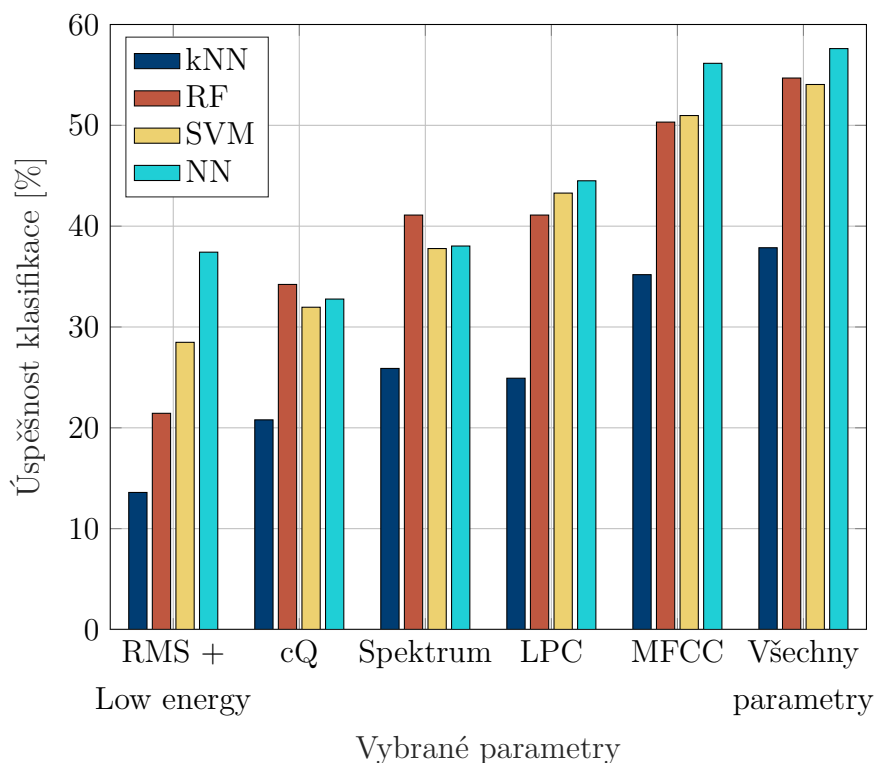
Jako absolutně nejlepší parametry ke klasifikaci hudebních nástrojů jsou Melovské spektrální koeficienty, úspěšnost klasifikace je při jejich použití velmi podobná úspěšnosti při použití všech zmiňovaných parametrů a liší se pouze o několik procent.

Tab. 7.2: Vliv použitých parametrů na úspěšnost různých klasifikátorů.

Vybrané parametry	úspěšnost klasifikace [%]			
	kNN	RF	SVM	NN
RMS + nízká energie	13.59	21.44	28.48	<b>37.42</b>
cQ	20.79	<b>34.22</b>	31.96	32.77
Spektrální parametry	25.89	<b>41.1</b>	37.78	38.03
LPC	24.92	41.1	43.28	<b>44.5</b>
MFCC	35.19	50.32	50.97	<b>56.15</b>
Všechny parametry	37.86	54.69	54.05	<b>57.61</b>

### 7.1.3 Testování dvojic nástrojů

V této části testování byla provedena klasifikace všech 55 dvojic 11 nástrojů s využitím neuronové sítě. Klasifikátor s NN byl vybrán díky své univerzálnosti a dobrým výsledkům v předchozích testech. Tabulka 7.3 obsahuje úspěšnosti testování těchto dvojic. Pro lepší přehled jsou výsledky s úspěšností nad 90 % označeny tučně, výsledky s úspěšností pod 75 % oranžovou barvou a výsledky s úspěšností pod 50 % červenou barvou. Nástroje náležící k uvedeným zkratkám jsou vypsané v tabulce 4.1.



Obr. 7.2: Vliv použitých parametrů na úspěšnost různých klasifikátorů.

Velmi nízkou úspěšnost vykazují výsledky klasifikace dechových nástrojů (saxofon, flétna, klarinet), které se jako jediné pohybují pod 50%. Tento výsledek není překvapivý zejména u záměny klarinetu a saxofonu. Oba zmiňované nástroje patří do skupiny jednoplátkových dřevěných dechových nástrojů, jejich zvuk je tedy tvořen stejným způsobem. Nízkou úspěšnost klasifikace s různými nástroji mají také klávesy. Tento výsledek může být způsoben množstvím zvukových rejstříků, kterými tento nástroj disponuje. Celkem překvapivá je vysoká úspěšnost při rozlišování strunných nástrojů, zejména violoncella a houslí, která se pohybuje nad 95%.

#### 7.1.4 Výsledky klasifikace nahrávek obsahujících jeden nástroj

Poslední část je zaměřena na detailní popis výsledků klasifikátoru využívajícího neuronovou síť k rozlišení všech jedenácti nástrojů. Využity jsou také všechny parametry vyfiltrované pomocí metod mRMR a SFFS.

K evaluaci klasifikátoru byly použity následující metriky: sensitivita, preciznost a F1 skóre. Sensitivita vyjadřuje, kolik testovacích nahrávek, které by měly být zařazeny do určité třídy, bylo do této třídy opravdu zařazeno:

$$sensitivita = \frac{TP}{TP + FN}. \quad (7.2)$$

Tab. 7.3: Úspěšnost klasifikace dvojic nástrojů, výsledky jsou uvedeny v %.

	vio	cel	tru	gac	voi	gel	pia	org	cla	flu	sax
vio	–	<b>95,77</b>	<b>95,38</b>	<b>90,53</b>	<b>90,09</b>	<b>91,11</b>	84,51	<b>72,32</b>	<b>68,83</b>	84,55	<b>90,00</b>
cel	<b>95,77</b>	–	<b>92,86</b>	<b>92,72</b>	<b>93,58</b>	<b>91,42</b>	<b>94,19</b>	<b>65,33</b>	<b>95,00</b>	79,45	<b>95,65</b>
tru	<b>95,38</b>	<b>92,86</b>	–	<b>90,50</b>	<b>90,00</b>	86,78	81,36	<b>62,32</b>	<b>55,88</b>	86,57	88,24
gac	<b>90,53</b>	<b>92,72</b>	<b>90,50</b>	–	88,25	83,95	86,05	82,19	<b>93,40</b>	<b>93,06</b>	86,15
voi	<b>90,09</b>	<b>93,58</b>	<b>90,00</b>	88,25	–	82,42	84,65	81,01	87,42	89,25	88,07
gel	<b>91,11</b>	<b>91,42</b>	86,78	83,95	82,42	–	86,74	77,01	<b>92,05</b>	<b>94,49</b>	87,84
pia	84,51	<b>94,19</b>	81,36	86,05	84,65	86,74	–	<b>74,81</b>	82,29	<b>73,89</b>	76,58
org	<b>72,32</b>	<b>65,33</b>	<b>62,32</b>	82,19	81,01	77,01	<b>74,81</b>	–	<b>58,02</b>	75,44	<b>62,50</b>
cla	<b>68,83</b>	<b>95,00</b>	<b>55,88</b>	<b>93,40</b>	87,42	<b>92,05</b>	82,29	<b>58,02</b>	–	<b>30,38</b>	<b>27,59</b>
flu	84,55	79,45	86,57	<b>93,06</b>	89,25	<b>94,49</b>	<b>73,89</b>	75,44	<b>30,38</b>	–	<b>45,16</b>
sax	<b>90,00</b>	<b>95,65</b>	88,24	86,15	88,07	87,84	76,58	<b>62,50</b>	<b>27,59</b>	<b>45,16</b>	–

Preciznost vyjadřuje, v kolika případech je třída označená klasifikátorem skutečně touto třídou:

$$preciznost = \frac{TP}{TP + FP}. \quad (7.3)$$

$TP$  vyjadřuje počet správných určení třídy,  $FP$  počet případů kdy je jiná třída označena za danou třídu a  $FN$  je počet případů z dané třídy, které jsou označeny jako třída jiná.  $F1$  skóre potom vychází ze senzitivity a preciznosti a používá se ke zobrazení jakéhosi kompromisu mezi těmito dvěma metrikami:

$$F1 = 2 \frac{preciznost \cdot senzitivita}{preciznost + senzitivita} \quad (7.4)$$

Výsledky klasifikace jsou znázorněny v matici záměn v tabulce 7.4, kde **SN** označuje skutečný nástroj a **PN** predikovaný nástroj označený klasifikátorem. Vypočítané metriky se nachází v tabulce 7.5. Celková úspěšnost této klasifikace je 57,61 %.

Výsledky ukazují, že klasifikátor nejlépe rozlišuje housle, obě kytary, zpěv a piano. Na druhou stranu má velký problém s rozlišením dechových nástrojů a elektrických kláves, což potvrzují i výsledky předchozí klasifikace dvojic. Nejméně přesné výsledky vykazuje klasifikace saxofonu, který nebyl správně identifikován ani jednou ze šesti nahrávek. Překvapivý je fakt, že byl v pěti případech označen za akustickou kytaru, která tvoří tón rozdílným způsobem a člověkem jsou tyto nástroje velmi lehce rozpoznatelné.

Obecně nízkou úspěšnost klasifikace můžeme pozorovat i u ostatních dechových nástrojů, a to hlavně dřevěných. Například flétna byla v naprosté většině případů označena za klarinet. Klarinet zase za trubku nebo violoncello. Trumpeta byla

správně rozpoznána ve většině případů, často za ní ale byly označovány jiné nástroje. V tabulce 7.5 tak u trubky pozorujeme velký rozdíl mezi sensitivitou a precizností.

Tab. 7.4: Matice záměn nástrojů.

PN \ SN	vio	cel	tru	gac	voi	gel	pia	org	cla	flu	sax
vio	44	1	1	1	0	0	0	0	2	3	2
cel	0	14	0	2	0	1	0	0	0	0	0
tru	0	0	8	0	0	1	0	1	0	0	1
gac	11	12	3	128	2	10	6	13	2	0	2
voi	4	3	2	15	193	20	10	28	2	0	2
gel	2	11	12	1	8	138	8	10	0	0	6
pia	4	9	32	16	5	9	187	27	14	9	15
org	1	4	2	9	5	12	8	11	6	0	0
cla	7	0	6	0	0	1	2	1	2	0	4
flu	0	1	2	1	1	1	8	2	30	4	6
sax	0	0	1	5	0	0	0	0	0	0	0

Tab. 7.5: Výsledky klasifikace nástrojů.

nástroj	testovací nahrávky	sensitivita	preciznost	F1 skóre
vio	54	0.8148	0.6027	0.6929
cel	17	0.8235	0.2545	0.3889
tru	11	0.7273	0.1159	0.2000
gac	189	0.6772	0.6465	0.6615
voi	279	0.6918	0.9019	0.7830
gel	216	0.6389	0.7150	0.6748
pia	327	0.5719	0.8166	0.6727
org	58	0.1897	0.1183	0.1457
cla	23	0.0870	0.0345	0.0494
flu	56	0.0714	0.2500	0.1111
sax	6	0.0000	0.0000	0.0000



## 7.2 Testování nahrávek obsahujících více nástrojů

Pro multi-label klasifikaci byl využit model popsany v kapitole 6.7, přičemž testovací nahrávky obsahovaly 2 až 4 různé hudební nástroje z 11 možných. Model byl natrénován stejnými daty, jako v případě klasifikace nahrávek s jedním nástrojem. Každá trénovací nahrávka tedy obsahovala pouze jeden nástroj.

V případě multi-label klasifikace je evaluace systému složitější. K evaluaci se v tomto případě může používat například *Exact Match Ratio – EMR*, jedná se o nejprísnejší metriku, která nebere ohled na skutečnost, že klasifikátor může mít i částečnou pravdu, kdy správně predikuje jen určitou část tříd z labelů. Tato metrika tedy výsledek považuje za správný pouze v případě, kdy jsou správně predikovány všechny labely, které do dané nahrávky patří (označeny jedničkou) a i ty, které do ní nepatří (označeny nulou). Nabývá hodnot od 0 do 1. V tabulce 7.6 je zobrazena ukázka výsledků klasifikace použité neuronové sítě pro prvních pět nahrávek z testovacího datasetu. Z výsledků vyplývá, že exact match kritérium splňuje pouze první nahrávka. Přitom je ale zřejmé, že je predikce dobrá i u dalších nahrávek.

Další používanou metrikou je *Hammingova ztráta (Hamming loss)* označována jako  $l_H$ . Ta se pro každou testovanou nahrávku vypočítá podle vzorce

$$l_H(\mathbf{y}, \mathbf{d}) = \frac{1}{m} \sum_{i=1}^m 1_{(y_i \neq d_i)}, \quad (7.5)$$

kde  $\mathbf{d}$  je vektor reálných labelů (žádaný výstup),  $\mathbf{y}$  je vektor predikovaných labelů a  $m$  je počet tříd. První se tedy spočítá počet labelů, pro které je predikce nesprávná a poté se normalizuje [8]. Výsledkem pro všechna testovací data je potom průměrná hodnota  $l_H$ , která vyjadřuje poměr nesprávně určených labelů k celkovému počtu labelů, její hodnota by tedy měla být co nejmenší.

V tabulce 7.7 a grafu 7.3 jsou zobrazeny hodnoty  $l_H$  a *EMR* v závislosti na počtu nástrojů v testovaných nahrávkách. Pro referenci byl tento typ sítě testován i na nahrávkách obsahujících pouze jeden nástroj. Tento výsledek je uveden v prvním řádku a zobrazen prvním sloupcem.

Je zřejmé, že i pro nahrávky s jedním nástrojem je úspěšnost klasifikace sítě nižší, než v předchozím případě. To je způsobeno skutečností, že systém neví, pro kolik nástrojů se má rozhodovat a tím pádem bere v potaz všechny možné kombinace se všemi počty nástrojů.

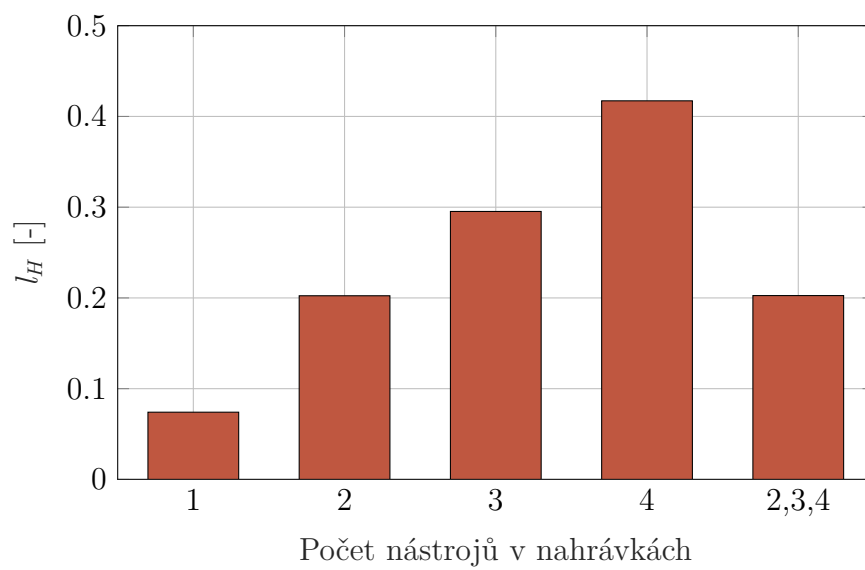
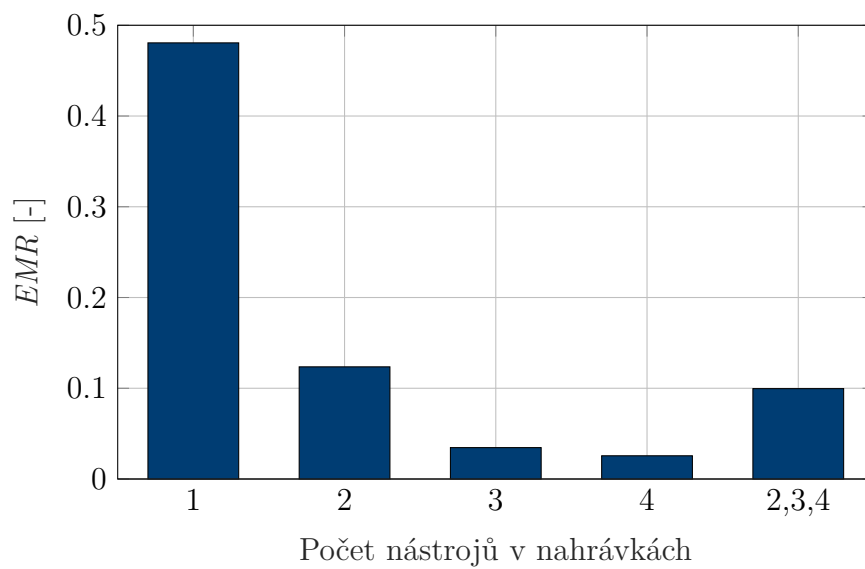
U nahrávek s vyšším počtem nástrojů se přesnost sítě prudce snižuje, což je nejvíce viditelné na hodnotě *EMR*, která se s rostoucím počtem zhoršuje rychleji, než  $l_H$  (pokles *EMR* má exponenciální charakter, zatímco  $l_H$  roste spíše lineárně). To znamená, že má klasifikátor větší problémy s určením celkového počtu a druhu hudebních nástrojů. Často ale klasifikuje správně alespoň některý z přítomných nástrojů.

Tab. 7.6: Ukázka multi-label predikce, bílá místa reprezentují nuly.

	reálné labely										
indexy tříd	0	1	2	3	4	5	6	7	8	9	10
1. nahrávka					1						1
2. nahrávka				1	1						
3. nahrávka		1								1	
4. nahrávka				1	1						1
5. nahrávka					1						1
	predikované labely										
indexy tříd	0	1	2	3	4	5	6	7	8	9	10
1. nahrávka					1						1
2. nahrávka				1						1	
3. nahrávka		1						1		1	
4. nahrávka					1						1
5. nahrávka					1	1			1		1

Tab. 7.7: Vliv počtu nástrojů v nahrávkách na  $l_H$  a  $EMR$ .

Počet nástrojů	$EMR$	$l_H$
1	0.4806	0.0741
2	0.1236	0.2024
3	0.0346	0.2953
4	0.0256	0.4172
2,3,4	0.0996	0.2026



Obr. 7.3: Vliv počtu nástrojů v nahrávkách na  $l_H$  a  $EMR$ .

## 8 Závěr

Cílem práce bylo navrhnout a implementovat systém k identifikaci hudebních nástrojů z hudebních nahrávek pomocí MIR.

Na začátku práce je popsán vznik hudebního tónu a jeho vlastnosti dané hudebním nástrojem, ze kterého tón vychází. Stejná kapitola také pojednává o rozdělení hudebních nástrojů do kategorií podle vzniku tónu.

Druhá kapitola je zaměřena na příznaky používané obecně v MIR a výběr těch parametrů, které nesou relevantní informace o hudebních nástrojích. Mezi ně patří hlavně parametry popisující barvu zvuku, jako jsou MFCC, LPC a další. Kapitola se také zaměřuje na softwarové nástroje, které jsou vhodné k efektivní extrakci těchto parametrů z hudebních nahrávek.

Třetí kapitola pojednává o klasifikačních metodách, které jsou často používány v oblasti zpracování signálu. Zaměřuje se hlavně na klasifikátory s učitelem, jako jsou například neuronové sítě, které jsou v dnešní době velmi často používány.

Další kapitola popisuje dvě obsáhlé databáze hudebních nahrávek, které se často používají pro MIR. K trénování a testování implementovaného systému byla vybrána databáze *IRMAS*, která obsahuje 11 druhů hudebních nástrojů, jejichž nahrávky pocházejí přímo z hudebních skladeb.

Pátá kapitola shrnuje poznatky již uskutečněných zahraničních prací a porovnává jejich výsledky s ohledem na použité parametry, datasety a klasifikační systémy.

Následující kapitola se zaměřuje na návrh a realizaci popsaného systému. Popisuje přípravu hudebních souborů jejich předzpracování a samotnou parametrizaci. Je v ní popsána tvorba příznakového vektoru, jeho redukce pomocí metod mRMR a SFFS a následná tvorba databáze ze získaných dat. Kapitola dále obsahuje popis ladění jednotlivých použitých klasifikátorů, které byly implementovány v prostředí Python 3.6. Zaměřuje se hlavně na neuronové sítě, které byly použity k single-label i multi-label klasifikaci.

Poslední kapitola obsahuje výsledky všech těchto klasifikátorů při klasifikaci nástrojů z nahrávek obsažených v testovacím setu *IRMAS*. Při klasifikaci všech nástrojů v nahrávkách s jedním obsaženým nástrojem, byly výsledky v porovnání s pracemi z kapitoly 5 na podobné úrovni. Nejlepší výsledky vykazovala neuronová síť, která identifikovala nástroje s úspěšností 57,61 %. Nejhůře si vedl kNN klasifikátor. I když se výsledky některých zahraničních akademiků pohybovaly i nad úspěšností 90 %, je potřeba brát v úvahu počet klasifikovaných nástrojů. Z výsledků je zřejmé, že se úspěšnost klasifikace s rostoucím počtem tříd prudce snižuje. Dalším důležitým faktorem je použitý dataset. Mnoho prací využilo databázi *The University of Iowa Musical Instrumental Samples*, která obsahuje pouze nahrávky izolovaných tónů pořízené v bezodrazové komoře, což je výhoda při následné klasifikaci testo-

vané na podobně izolovaných nahrávkách. V neposlední řadě je velmi důležitý i výběr parametrů. Při testování jsme došli k výsledku, že nejvhodnějšími parametry jsou MFCC koeficienty, jejichž využití vykazovalo téměř stejnou úspěšnost identifikace nástrojů, jako klasifikace se všemi vypočítanými parametry.

Při klasifikaci nástrojů, které se v nahrávkách nevyskytují samy, jsme došli k opo-  
znání horším výsledkům. Při testování tohoto systému na nahrávkách s jedním ná-  
strojem byla úspěšnost menší o asi 10 %, s přidáním nástrojů však prudce klesala.  
Tento trend byl zachycen s použitím speciálních metrik pro multi-label klasifikaci.

Nepřesnost některých klasifikátorů (hlavně kNN) může být způsobena špat-  
ným nastavením na validačním setu, který se může příliš lišit od setu testovacího.  
Výsledky této práce by tedy bylo možné zlepšit použitím obsáhlejšího datasetu, vý-  
běrem většího počtu parametrů a lepším vyladěním hyperparametrů klasifikátorů,  
a to hlavně neuronových sítí, které mohou být velmi přesné díky své variabilitě. Ta  
má ale za následek časově náročné ladění a dlouhé trénovací časy.

## Literatura

- [1] AZARLOO, A. a F. FAROKHI. Automatic Musical Instrument Recognition Using K-NN and MLP Neural Networks. In: *Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on* [online]. Phuket, Thailand: IEEE, 2012, s. 289-294 [cit. 2018-12-01]. DOI: 10.1109/CICSyN.2012.61. ISBN 9781467326407. Dostupné z: <https://ieeexplore.ieee.org/document/6274395>
- [2] BHALKE, D., C. RAO a D. BORMANE. Automatic musical instrument classification using fractional fourier transform based- MFCC features and counter propagation neural network. *Journal of Intelligent Information Systems* [online]. New York: Springer US, 2016, **46**(3), 425-446 [cit. 2018-11-20]. DOI: 10.1007/s10844-015-0360-9. ISSN 0925-9902. Dostupné z: <https://link.springer.com/article/10.1007/s10844-015-0360-9>
- [3] BOSCH, J, J JANER, F FUHRMANN a P HERRERA. *A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals* [online]. Porto, Portugalsko: 13th International Society for Music Information Retrieval Conference (ISMIR 2012), 2012, 2012 [cit. 2018-12-11]. Dostupné z: <http://mtg.upf.edu/system/files/publications/Bosch-ISMIR2012.pdf>
- [4] Music Information Retrieval. SCHREIBMAN, Susan, Ray SIEMENS a John UNSWORTH. *A new companion to digital humanities* [online]. 2. vydání. Chichester, West Sussex, UK: Wiley/Blackwell, 2016, s. 213—224 [cit. 2018-11-05]. ISBN 9781118680599.
- [5] BUSHAEV, Vitaly. Adam-: -latest trends in deep learning optimization. *Towards Data Science* [online]. 22. 10. 2018 [cit. 2019-05-13]. Dostupné z: <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>
- [6] CANNAM, Chris, Christian LANDONE a Mark SANDLER. *Sonic Visualiser: An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files* [online]. 2010 [cit. 2018-12-02]. Dostupné z: <https://www.sonicvisualiser.org/sv2010.pdf>
- [7] ČERNOCKÝ, Jan. *Zpracování řečových signálů: studijní opora*. Brno, 2006. Ústav počítačové grafiky a multimédií, Fakulta informačních technologií, Vysoké učení technické v Brně.

- [8] DESTERCKE, Sébastien. Multilabel Prediction with Probability Sets: The Hamming Loss Case. *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2014)* [online]. France, 2014, , 496-505 [cit. 2019-05-14]. Dostupné z: <https://hal.archives-ouvertes.fr/hal-01044994/document>
- [9] DIMENT, Aleksandr, Toni HEITTOLA a Tuomas VIRTANEN. Semi-supervised learning for musical instrument recognition. In: *European Signal Processing Conference* [online]. European Signal Processing Conference, EUSIPCO, 2013 [cit. 2018-11-05]. ISBN 9780992862602. ISSN 22195491.
- [10] FUHRMANN, Ferdinand. *Automatic musical instrument recognition from polyphonic music audio signals*. Barcelona, Španělsko, 2012. Disertační práce. Universitat Pompeu Fabra. Vedoucí práce Dr. Xavier Serra.
- [11] GAN, John Q., Bashar AWWAD SHIEKH HASAN a Chun Sing Louis TSUI. A filter-dominating hybrid sequential forward floating search method for feature subset selection in high-dimensional space. *International Journal of Machine Learning and Cybernetics*. 2014, **5**(3), 413-423. DOI: 10.1007/s13042-012-0139-z. ISSN 1868-8071. Dostupné také z: <http://link.springer.com/10.1007/s13042-012-0139-z>
- [12] GAO, Bin, W. WOO a S. DLAY. Unsupervised Single-Channel Separation of Nonstationary Signals Using Gammatone Filterbank and Itakura-Saito Non-negative Matrix Two-Dimensional Factorizations. *IEEE Transactions on Circuits and Systems I: Regular Papers* [online]. 2013, **60**(3), 662-675 [cit. 2018-12-13]. DOI: 10.1109/TCSI.2012.2215735. ISSN 1549-8328. Dostupné z: <http://search.proquest.com/docview/1365137821/>
- [13] HANCHUAN PENG, FUHUI LONG a C. DING. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* [online]. USA: IEEE, 2005, **27**(8), 1226-1238 [cit. 2018-12-11]. DOI: 10.1109/T-PAMI.2005.159. ISSN 0162-8828. Dostupné z: <https://ieeexplore.ieee.org/document/1453511>
- [14] HOLČÍK, Jiří. *Analýza a klasifikace dat*. Brno: Akademické nakladatelství CERM, 2012. ISBN 978-80-7204-793-2.
- [15] HONZÍK, Petr. *Strojové učení*. Brno, 2006. Skripta. VUT FEKT.

- [16] CHIA AI, Ooi, M. HARIHARAN, Sazali YAACOB a Lim SIN CHEE. Classification of speech dysfluencies with MFCC and LPCC features. *Expert Systems with Applications* [online]. 2012, **39**(2), 2157-2165 [cit. 2018-11-10]. DOI: 10.1016/j.eswa.2011.07.065. ISSN 09574174. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S095741741101027X>
- [17] JIRSÍK, Václav. *MUIN — Umělá inteligence: Vícevrstvá neuronová síť s algoritmem učení backpropagation*. Brno, 2013. Přednáška. VUT FEKT.
- [18] KAMINSKYJ, Ian a Tadeusz CZASZEJKO. Automatic Recognition of Isolated Monophonic Musical Instrument Sounds using k NNC. *Journal of Intelligent Information Systems* [online]. Boston: Kluwer Academic Publishers, 2005, **24**(2), 199-221 [cit. 2018-11-20]. DOI: 10.1007/s10844-005-0323-7. ISSN 0925-9902. Dostupné z: <https://link.springer.com/article/10.1007/s10844-005-0323-7>
- [19] KARPATY, Andrej. Convolutional Neural Networks for Visual Recognition. *CS231n* [online]. Stanford, California, USA: Stanford University, 2019 [cit. 2019-05-12]. Dostupné z: <http://cs231n.github.io/neural-networks-1>
- [20] KAZI, F.I. a D.G. BHALKE. Musical instrument classification using higher order spectra and MFCC. *2015 International Conference on Pervasive Computing (ICPC)* [online]. IEEE, 2015, 2015, , 1-6 [cit. 2018-11-27]. DOI: 10.1109/PERVASIVE.2015.7087048. ISBN 978-1-4799-6272-3. Dostupné z: <http://ieeexplore.ieee.org/document/7087048/>
- [21] KOTHE, R.S., D.G. BHALKE a P.P. GUTAL. Musical instrument recognition using k-nearest neighbour and Support Vector Machine. In: *Advances in Electronics, Communication and Computer Technology (ICAECCT), 2016 IEEE International Conference on* [online]. Pune, India: IEEE, 2016, s. 308-313 [cit. 2018-11-18]. DOI: 10.1109/ICAECCT.2016.7942604. Dostupné z: <https://ieeexplore.ieee.org/document/7942604>
- [22] KURFÜRST, Pavel. *Hudební nástroje*. Praha: Togga, 2002. ISBN 80-902-9121-X.
- [23] LARTILLOT, Olivier. *MIRtoolbox 1.7.1: User-s Manual*. Oslo, Norsko: University of Oslo, Department of Musicology, 2018.
- [24] LERCH, Alexander. *An introduction to Audio content Analysis: Applications in Signal Processing and Music Informatics*. Hoboken, New Jersey: Wiley, 2012. ISBN 978-1-118-26682-3.



- [25] LYONS, James. Mel Frequency Cepstral Coefficient (MFCC) tutorial. *Practical Cryptography* [online]. 2017 [cit. 2018-11-18]. Dostupné z: <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>
- [26] OZEROV, A. a E. VINCENT. A General Flexible Framework for the Handling of Prior Information in Audio Source Separation. *Audio, Speech, and Language Processing, IEEE Transactions on* [online]. USA: IEEE, 2012, **20**(4), 1118-1133 [cit. 2018-12-13]. DOI: 10.1109/TASL.2011.2172425. ISSN 1558-7916. Dostupné z: <http://hal.inria.fr/hal-00626962/>
- [27] PARDO, Bryan. Music information retrieval. *Communications of the ACM* [online]. 2006, **49**(8) [cit. 2018-11-11]. DOI: 10.1145/1145287.1145309. ISSN 0001-0782. Dostupné z: <http://search.proquest.com/docview/28940689/>
- [28] PEETERS, Geoffroy. *A large set of audio features for sound description (similarity and classification) in the CUIDADO project* [online]. Paris, France, 2004 [cit. 2018-11-15]. Dostupné z: [http://recherche.ircam.fr/anasy/peeters/ARTICLES/Peeters\\_2003\\_cuidadoaudiofeatures.pdf](http://recherche.ircam.fr/anasy/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf). IRCAM, Analysis/Synthesis Team.
- [29] REYNOLDS, Douglas A., Thomas F. QUATIERI a Robert B. DUNN. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing* [online]. Elsevier, 2000, **10**(1-3), 19-41 [cit. 2018-11-25]. DOI: 10.1006/dspr.1999.0361. ISSN 1051-2004. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1051200499903615>
- [30] SANKAYE, Satish, Suresh MEHROTRA a U TANDON. Indian Musical Instrument Recognition using Modified LPC Features. *International Journal of Computer Applications* [online]. New York: Foundation of Computer Science, 2015, **122**(13) [cit. 2018-11-11]. DOI: 10.5120/21758-4991. ISSN 09758887. Dostupné z: <http://search.proquest.com/docview/1698432584/>
- [31] SCHEDL, Markus, Emilia G-MEZ GUTIÉRREZ a Julián URBANO. *Music information retrieval: recent developments and applications* [online]. Now Publishers, 2014, **8**(2-3) [cit. 2018-11-15]. DOI: 10.1561/15000000042. ISSN 1554-0669. Dostupné z: <http://hdl.handle.net/10230/27565>
- [32] SCHWRAZ, Petr. *Systémy zpracování řeči: Umělé neuronové sítě a Support Vector Machines*. Brno, 2009. Studijní opora. Fakulta informačních technologií, Vysoké učení technické v Brně.

- [33] SYROVÝ, Václav. *Hudební akustika*. 3., dopl. vyd. V Praze: Akademie múzických umění, 2013. Akustická knihovna Zvukového studia Hudební fakulty AMU. ISBN 978-80-7331-297-8.
- [34] ZHANG, Min-ling a Zhi-hua ZHOU. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transactions on Knowledge and Data Engineering* [online]. IEEE, 2006, **18**(10), 1338-1351 [cit. 2019-05-13]. DOI: 10.1109/TKDE.2006.162. ISSN 1041-4347. Dostupné z: [ieeexplore.ieee.org/document/1683770](http://ieeexplore.ieee.org/document/1683770)
- [35] *Musical Instrument Samples: Electronic Music Studios* [online]. Iowa City, USA: University of Iowa [cit. 2018-12-11]. Dostupné z: <http://theremin.music.uiowa.edu/MIS.html>

# Seznam symbolů, veličin a zkratek

<b>ADSR</b>	Attack, Decay, Sustain, Release
<b>API</b>	- Application programming interface – rozhraní pro programování aplikací Attack, Decay, Sustain, Release
<b>CQT</b>	Constant Q Transform – konstantní Q transformace
<b>DO</b>	Drop out
<b>DT</b>	Disizion Tree – rozhodovací strom
<b>EMR</b>	Exact Match Ratio
<b>FASST</b>	Flexible Audio Source Separation Toolbox
<b>FFT</b>	Fast Fourier Transform – rychlá Fourierova transformace
<b>FN</b>	False Negative
<b>FP</b>	False Positive
<b>GMM</b>	Gaussian Mixture Models
<b>IRMAS</b>	Dataset for Intrument Recognition in Musical Audio Signals
<b>kNN</b>	k-Nearest Neighbor – k-nejbližší soused
<b><math>l_H</math></b>	Hamming Loss – Hammingova ztráta
<b>LPC</b>	Linear Prediction Coefficients – lineární predikční koeficienty
<b>MFCC</b>	Mel Frequency Cepstral Coefficients – melovské keprstrální koeficienty
<b>MIR</b>	Music Information Retrieval
<b>MIS</b>	Musical Instrument Samples
<b>MLP NN</b>	Multilayer Perceptronal Neural Network – vícevrstvá perceptronová neuronová síť
<b>MRMR</b>	Minimum Redundancy Maximum Relevance
<b>NEST</b>	New and Emerging Science and Technology
<b>NN</b>	Neural Network – neuronová síť
<b>PCA</b>	Principal Component Analysis – analýza hlavních komponent
<b>ReLU</b>	Rectified Linear Unit
<b>RF</b>	Random Forest – náhodný les
<b>RMS</b>	Root Mean Square – efektivní hodnota
<b>SF</b>	Spectral Flux – spektrální tok
<b>SK</b>	Spectral Kurtosis – spektrální špičatost
<b>SVM</b>	Support Vector Machine – podpůrné vektory
<b>TP</b>	True Positive
<b>USCSS</b>	Unsupervised Single-Channel Separation of Nonstationary Signals
<b>ZCR</b>	Zero Crossing Rate – počet průchodů nulovou úrovní

# Seznam příloh

A	Obsah přiloženého CD	68
B	Seznam parametrů získaných pomocí mRMR	69
C	Seznam parametrů získaných pomocí SFFS	71

# A Obsah přiloženého CD

Přiložené CD obsahuje samotný text práce ve formátu pdf, skripty psané v prostředí Matlab 2015a a skripty psané v prostředí Python 3.6.

Skripty `movefiles.m`, a `preprocessing.m` slouží k přípravě nahrávek. Skripty `showtransforms.m` a `getfeatures.m` využívají program Sonic Annotátor a příslušné *vamp plugins*, které musí být v pc nainstalovány.

Skripty pro jazyk Python 3.6 označené jako `classifier` značí implementaci jednotlivých klasifikátorů a SFFS. Skript `functions.py` obsahuje funkci na nahrání dat ve formátu popsaném v této práci, funkci na rozdělení dat k validaci a dále funkce na zápis výsledku do csv souboru, vyfiltrování dat pomocí SFFS, evaluaci, kalkulaci vah a výběr jednotlivých hudebních nástrojů. `bp_mll_keras.py` obsahuje funkci pro výpočet chyby při multi-label klasifikaci. Pro jejich funkcionalitu je nutné nainstalovat knihovny uvedené v hlavičkách. Knihovna `tensorflow-gpu` navíc vyžaduje instalaci *Nvidia CUDA* ovladače.

```
/ ..... kořenový adresář přiloženého CD
├── Karnik.R_DP.pdf ..... text práce
├── Skripty
│   ├── Python ..... skripty psané v prostředí Python 3.6
│   │   ├── SFFS.py
│   │   ├── functions.py
│   │   ├── kNN_classifier.py
│   │   ├── SVM_classifier.py
│   │   ├── NN_sl_classifier.py
│   │   ├── NN_ml_classifier.py
│   │   ├── RF_classifier.py
│   │   └── bp_mll_keras.py
│   └── Matlab ..... skripty psané v prostředí Matlab 2015a
│       ├── movefiles.m
│       ├── preprocessing.m
│       ├── showtransforms.m
│       └── getfeatures.m
```

## B Seznam parametrů získaných pomocí mRMR

Tab. B.1: Seznam parametrů získaných pomocí mRMR: 1.–70. parametr (p. = percentil, q. = kvartil).

<b>pořadí</b>	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
<b>parametr</b>	s_kurtosis 1.p	RMS 1.q	MFCC 4 max	MFCC 18 std	s_rolloff 1.q	MFCC 9 99.p	cQ 40 max	LPC4 min	MFCC 4 std	cQ 2 max
<b>pořadí</b>	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
<b>parametr</b>	MFCC 3 min	MFCC 7 99.p	cQ 21 99.p	LPC 13 std	MFCC 5 99.p	cQ 38 99.p	LPC 5 99.p	MFCC 4 min	s_kurtosis std	s_kurtosis min
<b>pořadí</b>	21.	22.	23.	24.	25.	26.	27.	28.	29.	30.
<b>parametr</b>	MFCC 11 std	cQ 42 99.p	cQ 17 std	LPC 1 std	cQ 24 99.p	MFCC 6 99.p	s_rolloff 3.q	s_spread std	cQ 47 99.p	cQ 15 1.q
<b>pořadí</b>	31.	32.	33.	34.	35.	36.	37.	38.	39.	40.
<b>parametr</b>	LPC 3 99.p	MFCC 3 std	MFCC 12 max	LPC 13 1.q	cQ 19 max	MFCC 2 min	MFCC 3 max	MFCC 17 std	cQ 29 max	cQ 3 1.q
<b>pořadí</b>	41.	42.	43.	44.	45.	46.	47.	48.	49.	50.
<b>parametr</b>	LowE	cQ44 max	LPC 12 1.p	cQ 14 99.p	cQ 36 max	MFCC 1 3.q	MFCC 18 3.q	MFCC 4 1.p	s_spread 99.p	cQ 39 max
<b>pořadí</b>	51.	52.	53.	54.	55.	56.	57.	58.	59.	60.
<b>parametr</b>	MFCC 9 median	MFCC 10 std	s_kurtosis median	cQ 18 1.q	LPC 4 1.p	s_rolloff 1.p	MFCC 5 std	ZCR 99.p	cQ 46 std	MFCC 7 mean
<b>pořadí</b>	61.	62.	63.	64.	65.	66.	67.	68.	69.	70.
<b>parametr</b>	cQ 1 std	MFCC 16 std	MFCC 3 1.q	RMS max	LPC 6 1.q	cQ 27 max	cQ 22 max	s_rolloff meidan	MFCC 7 std	s_skewness min

Tab. B.2: Seznam parametrů získaných pomocí mRMR: 71.–150. parametr (p. = percentil, q. = kvartil).

<b>pořadí</b>	71.	72.	73.	74.	75.	76.	77.	78.	79.	80.
<b>parametr</b>	MFCC 1 std	MFCC 5 mean	cQ 20 max	MFCC 2 max	MFCC 4 1.q	MFCC 15 std	RMS 3.q	s_skewness 1.p	cQ 42 std	LPC 5 max
<b>pořadí</b>	81.	82.	83.	84.	85.	86.	87.	88.	89.	90.
<b>parametr</b>	cQ 48 max	cQ 10 1.q	MFCC 9 max	LPC 3 3.q	MFCC 6 max	MFCC 19 1.q	MFCC 11 99.p	LPC 1 1.q	cQ 4 std	MFCC 14 99.p
<b>pořadí</b>	91.	92.	93.	94.	95.	96.	97.	98.	99.	100.
<b>parametr</b>	MFCC 3 3.q	LPC 8 1.q	MFCC 17 1.q	MFCC 6 std	cQ 18 99.p	MFCC 5 max	LPC 7 max	MFCC 12 std	MFCC 4 99.p	s_rolloff 99.p
<b>pořadí</b>	101.	102.	103.	104.	105.	106.	107.	108.	109.	110.
<b>parametr</b>	cQ 14 1.q	cQ 45 max	MFCC 7 max	MFCC 1 max	cQ 34 max	MFCC 2 1.q	cQ 41 max	cQ 23 max	MFCC 2 std	cQ 1 1.q
<b>pořadí</b>	111.	112.	113.	114.	115.	116.	117.	118.	119.	120.
<b>parametr</b>	s_spread 3.q	cQ 37 max	cQ 12 std	MFCC 10 99.p	MFCC 3 1.p	LPC 5 3.q	MFCC 16 3.q	s_rolloff mean	cQ 40 99.p	MFCC 18 99.p
<b>pořadí</b>	121.	122.	123.	124.	125.	126.	127.	128.	129.	130.
<b>parametr</b>	MFCC 3 99.p	cQ 16 3.q	LPC 13 99.p	MFCC 8 99.p	cQ 1 99.p	MFCC 1 min	LPC 11 max	MFCC 6 median	ZCR max	MFCC 19 std
<b>pořadí</b>	131.	132.	133.	134.	135.	136.	137.	138.	139.	140.
<b>parametr</b>	cQ 19 std	cQ 31 max	LPC 1 min	MFCC 7 3.q	MFCC 9 3.q	LPC 3 max	cQ 38 std	s_kurtosis 1.q	cQ 47 std	cQ 3 3.q
<b>pořadí</b>	141.	142.	143.	144.	145.	146.	147.	148.	149.	150.
<b>parametr</b>	cQ 21 mean	LPC 1 max	MFCC 5 3.q	MFCC 17 99.p	MFCC 2 1.p	MFCC 4 median	MFCC 9 std	LPC 6 1.p	RMS median	cQ 39 99.p

## C Seznam parametrů získaných pomocí SFFS

Tab. C.1: Seznam parametrů získaných pomocí SFFS: 1.–70. parametr (p. = precentil, q. = kvartil).

<b>pořadí</b>	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
<b>parametr</b>	s_kurtosis 1.p	RMS 1.q	MFCC 4 max	MFCC 18 std	s_rolloff 1.q	MFCC 9 99.p	LPC 4 min	MFCC 4 std	cQ 2 max	MFCC 7 99.p
<b>pořadí</b>	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
<b>parametr</b>	LPC 13 std	cQ 38 99.p	LPC 5 99.p	MFCC 11 std	cQ 17 std	LPC 1 std	cQ 24 99.p	cQ 47 99.p	cQ 15 1.q	LPC 3 99.p
<b>pořadí</b>	21.	22.	23.	24.	25.	26.	27.	28.	29.	30.
<b>parametr</b>	MFCC 3 std	LPC 13 1.q	MFCC 2 min	MFCC 3 max	cQ 44 max	LPC 12 1.p	cQ 14 99.p	MFCC 1 3.q	MFCC 18 3.q	s_spread 99.p
<b>pořadí</b>	31.	32.	33.	34.	35.	36.	37.	38.	39.	40.
<b>parametr</b>	cQ 39 max	MFCC 9 median	MFCC 10 std	cQ 18 1.q	LPC 4 1.p	s_rolloff 1.p	ZCR 99.p	cQ 1 std	MFCC 16 std	MFCC 3 1.q
<b>pořadí</b>	41.	42.	43.	44.	45.	46.	47.	48.	49.	50.
<b>parametr</b>	LPC 6 1.q	cQ 27 max	cQ 22 max	MFCC 7 std	MFCC 5 mean	cQ 20 max	MFCC 2 max	MFCC 4 1.q	MFCC 15 std	RMS 3.q
<b>pořadí</b>	51.	52.	53.	54.	55.	56.	57.	58.	59.	60.
<b>parametr</b>	s_skewness 1.p	cQ 42 std	LPC 5 max	cQ 48 max	cQ 10 1.q	MFCC 9 max	LPC 3 3.q	MFCC 6 max	MFCC 19 1.q	MFCC 11 99.p
<b>pořadí</b>	61.	62.	63.	64.	65.	66.	67.	68.	69.	70.
<b>parametr</b>	cQ 4 std	MFCC 14 99.p	LPC 8 1.q	MFCC 6 std	MFCC 4 99.p	s_rolloff 99.p	cQ 45 max	MFCC 1 max	cQ 34 max	MFCC 2 1.q



Tab. C.2: Seznam parametrů získaných pomocí SFFS: 71.–100. parametr (p. = percentil, q. = kvartil).

<b>pořadí</b>	71.	72.	73.	74.	75.	76.	77.	78.	79.	80.
<b>parametr</b>	cQ 23 max	MFCC 2 std	cQ 1 1.q	s_spread 3.Q	cQ 37 max	cQ 12 std	MFCC 10 99.p	MFCC 3 1.p	LPC 5 3.q	s_rolloff mean
<b>pořadí</b>	81.	82.	83.	84.	85.	86.	87.	88.	89.	90.
<b>parametr</b>	cQ 40 99.p	MFCC 3 99.p	cQ 16 3.q	MFCC 8 99.p	cQ 1 99.p	MFCC 1 min	LPC 11 max	MFCC 6 median	MFCC 19 std	cQ 31 max
<b>pořadí</b>	91.	92.	93.	94.	95.	96.	97.	98.	99.	100.
<b>parametr</b>	MFCC 7 3.q	LPC 3 max	s_kurtosis 1.q	cQ 3 3.q	cQ 21 mean	MFCC 5 3.q	MFCC 4 median	LPC 6 1.p	RMS median	cQ 39 99.p