

České vysoké učení technické v Praze  
Fakulta elektrotechnická

Katedra elektromagnetického pole



# Zvýrazňování řečového signálu na bázi hlubokých neuronových sítí

BAKALÁŘSKÁ PRÁCE

Autor: Jiří Šmíd

Vedoucí: Doc. Ing. Petr Pollák, CSc.

Datum: Květen, 2023



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šmíd** Jméno: **Jiří** Osobní číslo: **491855**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra elektromagnetického pole**  
Studijní program: **Elektronika a komunikace**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Zvýrazňování řečového signálu na bázi hlubokých neuronových sítí**

Název bakalářské práce anglicky:

**Speech Enhancement Based on Deep Neural Networks**

Pokyny pro vypracování:

- Seznamte se s problematikou zvýrazňování řeči s užším zaměřením na techniky potlačování aditivního šumu a konvolučního zkreslení řeči využívající hluboké neuronové sítě (DNN).
- Na základě provedené rešerše dostupných řešení vyberte vhodnou DNN architekturu a nástroje použitelné pro implementaci systému zvýrazňování řeči.
- Z databázi na pracovišti vedoucího resp. z volně dostupných zdrojů shromážděte data potřebná pro trénování DNN systému pro zvýrazňování řeči i pro následnou evaluaci.
- Vybrané přístupy srovnajte a vyhodnoťte míru potlačení aditivního šumu na pozadí a míru zkreslení zvýrazněného řečového signálu v různých podmínkách.

Seznam doporučené literatury:

- [1] O. Novotný: Improving Robustness of Speaker Recognition using Discriminative Techniques. Brno, 2021. Disertační práce. FIT, VUT v Brně. 2021.
- [2] X. Huang, A. Acero, H.-W. Hon: Spoken Language Processing. Prentice Hall, 2001.
- [3] D. Yu, L. Deng. Automatic Speech Recognition. A Deep Learning Approach. Springer-Verlag London. 2015

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**doc. Ing. Petr Pollák, CSc. katedra teorie obvodů FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **06.02.2023**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce: **22.09.2024**

doc. Ing. Petr Pollák, CSc.  
podpis vedoucí(ho) práce

\_\_\_\_\_ podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_ Datum převzetí zadání

\_\_\_\_\_ Podpis studenta



## Prohlášení

Prohlašuji, že svou závěrečnou práci na téma *Zvýrazňování řečového signálu na bázi hlubokých neuronových sítí* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

V Praze 21. 5. 2023

.....

Jiří Šmíd



## **Poděkování**

Rád bych poděkoval Doc. Ing. Petru Pollákovi, CSc. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce.

Jiří Šmíd





*Název:*

## **Zvýrazňování řečového signálu na bázi hlubokých neuronových sítí**

*Autor:* Jiří Šmíd

*Studijní program:* Elektronika a komunikace

*Druh práce:* Bakalářská práce

*Vedoucí práce:* Doc. Ing. Petr Pollák, CSc.  
Katedra teorie obvodů

*Klíčová slova :* neuronové sítě, potlačení šumu, zvýraznění řeči

*Key words:* neural networks, speech enhancement,  
noise suppression in speech signals

*Abstrakt:* Předložená práce se zabývá využitím standardní dopředné neuronové sítě pro zvýrazňování řeči resp. potlačování šumu v řečových signálech. Teoretická část práce se zabývá vlivem šumů na charakteristiky řečových signálů, měřením úrovně šumu v signálu a jeho zkreslením. Také jsou zmíněny tradiční metody zvýraznění řečového signálu. Hlavní část práce je věnována využití neuronových sítí pro zvýrazňování řeči, zmíněny jsou obecné vlastnosti, principy trénování a typické struktury neuronových sítí, které se využívají pro úlohu zvýrazňování řeči. V další části je popsána příprava a zpracování dat pro trénování sítě a vlastní implementace celého systému. Funkčnost navrženého systému byla testována v experimentální části, v rámci jednotlivých experimentů jsou popsány použité struktury neuronových sítí, nastavení jejich hyperparametrů i použité příznaky na vstupu sítě. Dosažené výsledky potvrdily, že systémem s neuronovými sítěmi lze dosáhnout dobrého potlačení aditivního šumu v řečovém signálu při jeho malém zkreslení.

*Abstract (English):* The presented work deals with the possibility of using fully connected forward neural networks for speech enhancement (noise suppression) in speech signals. The theoretical part describes the influence of noises in speech signals, the measurement of the amount of noise in speech signals, and their distortion. This part also discusses traditional methods of speech enhancement. The main part is dedicated to the usage of neural networks for speech enhancement. It describes general features, DNN training methods, and typical structures of DNNs for speech enhancement. The following part focuses on the preparation of data used for training the neural network, as well as the implementation of the entire system. The functionality of the designed system was tested in the experimental part. For each experiment, the structures of DNNs used, along with their hyperparameter settings and data format, are described. The results of these experiments confirmed that it is possible to achieve good suppression of additive noises without significant distortion using the neural network system.



# Seznam použitých zkratek

DNN	Hluboká neuronová síť (z angl. <i>Deep Neural Network</i> )
OLA	Metoda sčítání přesahů (z angl. <i>Overlap and Add</i> )
SNR	Odstup signálu od šumu (z angl. <i>Signal to Noise Ratio</i> )
MSE	Střední kvadratická odchylka (z angl. <i>Mean Squared Error</i> )
VAD	Detektor řečové aktivity (z angl. <i>Voice Activity Detector</i> )
FFT	Rychlá Fourierova transformace (z angl. <i>Fast Fourier Transformation</i> )
STDFT	Krátkodobá, diskrétní Fourierova transformace (z angl. <i>Short Time Discrete Fourier transformation</i> )
PESQ	Percepční hodnocení kvality řeči (z angl. <i>Perceptual Evaluation of Speech Quality</i> )
WADA	Amplitudová distribuční analýza (z angl. <i>Waveform amplitude distribution analysis</i> )
CPU	Centrální procesorová jednotka (z angl. <i>Central Processing Unit</i> )
GPU	Grafický procesor/ grafická karta (z angl. <i>Graphics Processing Unit</i> )
TPU	Procesor pro učení modelů AI (z angl. <i>Tensor Processing Unit</i> )
MVN	Normalizace na nulovou střední hodnotu a jednotkový rozptyl (z angl. <i>Mean Variance Normalisation</i> )
LR	Rychlost učení (z angl. <i>Learning Rate</i> )
Acc	Přesnost predikce (z angl. <i>Accuracy</i> )
fs	Vzorkovací frekvence

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Zvýrazňování řeči</b>	<b>3</b>
2.1	Šumy v řečových signálech . . . . .	3
2.2	Měření řečového signálu . . . . .	3
2.2.1	SNR . . . . .	3
2.2.2	PESQ . . . . .	4
2.3	Metoda spektrálního odečítání . . . . .	6
2.4	Zvýrazňování řeči za pomoci neuronových sítí . . . . .	8
2.4.1	Neuronové sítě . . . . .	8
2.4.2	Učení neuronové sítě . . . . .	11
2.4.3	Optimalizační algoritmy . . . . .	12
2.5	Neuronová síť na bázi spektrálního odečítání . . . . .	13
<b>3</b>	<b>Implementace</b>	<b>15</b>
3.1	Implementace přípravné části . . . . .	16
3.2	Implementace vstupního streamu (pipeline) . . . . .	19
3.3	Implementace neuronové sítě . . . . .	21
<b>4</b>	<b>Experimenty a výsledky</b>	<b>23</b>
4.1	Databáze využité pro trénování . . . . .	24
4.2	Prvotní experimenty . . . . .	25
4.3	Experiment 1: 3 skryté vrstvy po 1500 neuronech a T1500+CAR . . . . .	25
4.3.1	Korelace SNR a WADA SNR . . . . .	27
4.3.2	Formát vstupních dat . . . . .	27
4.3.3	Velikost dávky . . . . .	29
4.3.4	Velikost okolí . . . . .	30
4.3.5	Promíchání dat . . . . .	31
4.3.6	Zhodnocení experimentu . . . . .	32
4.4	Experiment 2: 3 skryté vrstvy po 1500 neuronech a T1500+GEN_NOISES . . . . .	34
<b>5</b>	<b>Závěr</b>	<b>35</b>
	<b>Seznam obrázků</b>	<b>37</b>
	<b>Seznam tabulek</b>	<b>38</b>
	<b>Bibliografie</b>	<b>39</b>
	<b>Přílohy</b>	<b>41</b>
A	Ukázka WADA SNR a PESQ pro část testovacího datasetu k T1500+CAR . . . . .	41

# 1 Úvod

Aplikace hlasových technologií zaznamenávají v posledních deseti letech vzestup. Nejedná se pouze o přenos hlasu od jednoho účastníka k druhému běžnou mobilní sítí či přes VoIP (Voice over Internet Protocol) nebo přes aplikace na telefonování přes internet např. WhatsApp. Pomocí hlasu je možné ovládat chytrý telefon, vyhledávat v online vyhledávacích či dokonce ovládat svůj dům za pomoci domácího asistenta. Další využití nalézají hlasové technologie v komunikaci s poskytovateli některých služeb, např. s bankami, kdy se uplatňuje přirozený dialog s hlasovým chatbotem, který reaguje na řečené informace do telefonu.

Pro všechny tyto aplikace je velice důležitá kvalita vstupního řečového signálu. Jelikož v reálných podmínkách není možné signál snímat v akusticky čistém a bezodrazovém prostředí, je nutné realizovat minimalizaci vlivu aditivního šumu ve snímaném signálu (šumu prostředí), případně jeho konvolučního zkreslení (tj. odrazů/dozvuku) softwarově.

Standardní metody zvýraznění řeči jsou založené na spektrálním odečítání nebo v případě snímání řeči více mikrofony je možné realizovat tvarování směrové charakteristiky mikrofonním polem. V posledních letech zaznamenávají úspěchy metody zvýrazňování řeči za pomoci neuronových sítí. Oproti tradičním metodám zvýrazňování vykazují lepší míru potlačení šumů, a to statických i dynamických, které představují pro tradiční metody zásadní problém. První využití neuronových sítí pro zvýrazňování řeči se objevilo koncem 20. století. Větší rozšíření tohoto přístupu lze pozorovat v posledních letech, kdy se systémy na bázi neuronových sítí pro tento účel stávají široce používaným standardem. Využívají se sítě konvoluční, autoencodery, či dopředné několikavrstvé plně propojené.

Hlavní motivací této práce je analýza využití dopředných plně propojených hlubokých neuronových sítí (DNN - Deep Neural Network) pro zvýrazňování řečového signálu. Práce je členěna do tří hlavních kapitol, první se zabývá vlivem šumu na charakteristiky řečových signálů, jeho měřením a metodami jeho potlačení. Druhá kapitola se zabývá přípravou dat pro trénování DNN, dále pak implementací této části i implementací samotné sítě. Čtvrtá kapitola popisuje realizované experimenty a shrnutí dosažených výsledků. V závěrečné kapitole je kromě shrnutí výsledků prezentováno také možné pokračování této práce.



# 2 Zvýrazňování řeči

## 2.1 Šumy v řečových signálech

Ve řečových signálech rozlišujeme dva základní druhy rušení, a to aditivní a konvoluční. Aditivní šumy jsou takové, které vznikají přičtením šumu k signálu. Toto rušení má jiný zdroj než signál a může mít nejrůznější původ, nejčastější je však hluk v pozadí např. konverzace lidí, šumění větru, hluk výrobních strojů, atd.

Za konvoluční šumy je označováno zkreslení vniklé odrazy, dozvuky, či zkreslením v přenosových kanálech.

Ve zpracování signálů existují i další šumy např. kvantizační, ale ty nejsou pro tuto práci důležité.

## 2.2 Měření řečového signálu

Chceme-li hodnotit kvalitu řečového signálu, je třeba znát úroveň šumu, míru zkreslení signálu, či jejich odhady.

### 2.2.1 SNR

Jedním z důležitých kritérií pro posuzování kvality řečového signálu je SNR (signal to noise ratio) neboli odstup signálu od šumu. Uvažujme, že zaznamenaný řečový signál  $x$  je složen ze dvou částí, z řečové části (užitečné informace)  $s[n]$  a z šumu  $n[n]$ .

$$x[n] = s[n] + n[n] \quad (2.1)$$

SNR pak definujeme jako poměr mezi výkonem řeči  $P_s$  a mezi výkonem šumu  $P_n$  tj.:

$$SNR = \frac{P_s}{P_n} [-] \quad (2.2)$$

Je-li je k dispozici čistý ekvivalent zašuměného signálu je tento výpočet jednoduchý jelikož  $s[n]$  je právě tento čistý signál. A rušení  $n[n]$  je možné získat jako rozdíl mezi  $x[n]$  a  $s[n]$ . V praxi však mnohdy není  $s[n]$  k dispozici a je nutné přistoupit k některým metodám odhadu míry šumu v signálu  $x[n]$ . V tomto případě pak dostáváme odhad  $\hat{P}_n$ , a  $\hat{P}_s$  je nutno vyjádřit jako rozdíl mezi  $P_x$  a  $\hat{P}_n$ . Výsledný vztah pro SNR je následující:

$$SNR = \frac{P_x - \hat{P}_n}{\hat{P}_n} [-] \quad (2.3)$$

Mnohem častěji je však SNR vyjadřováno za pomoci hladinového vyjádření tj. v dB. To je možné vypočítat jako:

$$SNR_{dB} = 10 \cdot \log_{10} SNR [dB] \quad (2.4)$$

Výpočet SNR za pomoci rozdílu zašuměného a čistého signálu je při využití metod zvýrazňování řečového signálu nevýhodný a nevypovídající, jelikož upravený signál bude vždy mít jiný tvar než signál čistý. SNR pak v tomto případě nepopisuje odstup signálu od šumu, ale rozdílnost upraveného signálu od čistého, bez ohledu na množství šumu v upraveném signálu, či míry srozumitelnosti signálu. Z tohoto důvodu je v experimentální části, jako jedno z vyhodnocovacích kritérií využito tzv. WADA SNR (Waveform amplitude distribution analysis). Tento algoritmus vychází z předpokladu, že hustota pravděpodobnosti pro amplitudy řečových signálů i pro zašuměné může být aproximována, za pomoci gamma distribuční funkce. Dalším z předpokladů je statistická nezávislost signálu a šumu. Tvar gamma distribuční funkce ovlivňují dva parametry  $\alpha$  a  $\beta$ . Při určování SNR je ze signálu napočtená gamma distribuční funkce. Z této funkce je spočten parametr  $G_z$ , jenž má souvislost se SNR, jelikož je také funkcí poměru výkonů čistého a zašuměného signálu, který je možno vyjádřit za pomoci parametrů  $\alpha$  a  $\beta$  pro čistý signál a šum. Z parametrů  $\alpha$  a  $\beta$  je tedy možné spočítat  $G_z$  a z něj přímo SNR.[1]

### 2.2.2 PESQ

PESQ (Perceptual Evaluation of Speech Quality) je další z důležitých metrik pro měření kvality řečových signálů. Prvotní použití tohoto indexu bylo v telekomunikaci pro evaluaci kvality přenášené řeči. Algoritmus jeho výpočtu je založen na percepčním modelu lidského slyšení a zahrnuje i citlivost lidského ucha na různé druhy zkreslení a schopnost mozku je kompenzovat. Algoritmus porovnává zkreslený řečový signál s originálním a skládá se z následujících fází:

1. Preprocessing: V této fázi jsou oba signály zbaveny přebytečných informací, které by mohly mít negativní vliv na vyhodnocení. Mezi tyto přebytečné informace patří části signálu, které neobsahují promluvy. Dále jsou oba signály normovány na stejné úrovni.



2. Analýza: Oba signály jsou následně filtrovány kaskádou filtrů, které simulují frekvenční odezvu lidského ucha a mají za cíl zachytit ty části spekter, které jsou nejpodstatnější pro lidské vnímání řeči.
3. Mapování: V této poslední fázi jsou rozdíly mezi oběma signály převedeny za pomoci nelineárních transformací (vzájemná korelace a další) na výstup v podobě PESQ indexu.

PESQ index může nabývat hodnot od -0.5 do 4.5, přičemž vyšší hodnota indexu odpovídá lepší kvalitě řečového signálu. Nevýhodou tohoto indexu je jeho závislost na jazyce měřené promluvy. Ve zvýrazňování řeči je tedy nutné porovnávat PESQ indexy původního signálu a signálu upraveného.

## 2.3 Metoda spektrálního odečítání

Metoda spektrálního odečítání je jednou z nejstarších a velmi často používaných metod zvýrazňování řečových signálů. Při použití této metody je předpokladem, že většina ruchů v řečových signálech je způsobena aditivními šumy tj. předpokládá že zašuměný signál je složen z řečového signálu a šumu viz rovnice 2.1

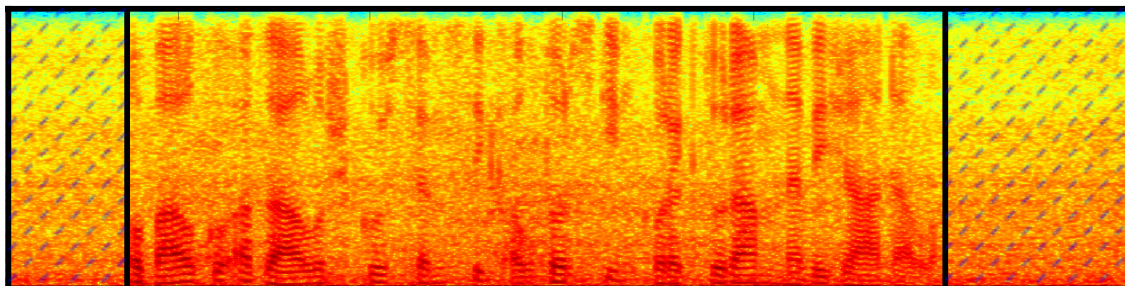
kde  $s[n]$  je řečový signál a  $n[n]$  je aditivním šumem. Předpokládejme, že šum má nulovou střední hodnotu a není korelovaný z řečovým signálem. Jelikož řečový signál není stacionární a není časově invariantní, bývá zašuměný signál zpracován po segmentech. Amplitudová spektra, po aplikaci STDFFT (Short Time Discrete Fourier Transform), těchto segmentů můžeme zapsat jako:

$$|X_l[k]| = |S_l[k]| + |N_l[k]|, \quad (2.5)$$

kde  $k$  reprezentuje pořadí spektrální čáry v diskretním spektru a  $l$  pořadí jednotlivých segmentů. Z tohoto vztahu je patrné, že je možné odhadnout spektra řečového signálu jako:

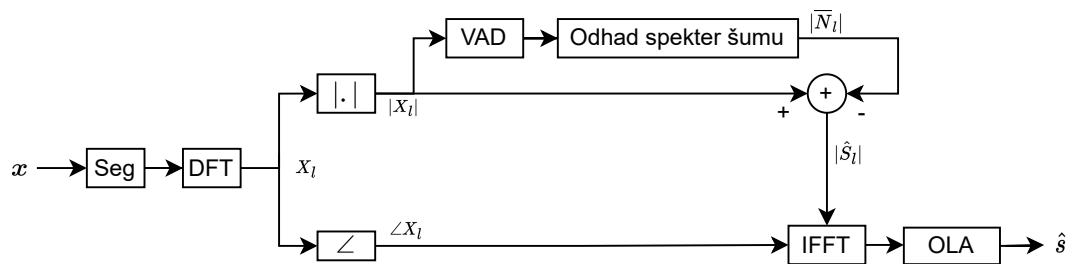
$$|\hat{S}_l[k]| = |X_l[k]| - |\bar{N}_l[k]|, \quad (2.6)$$

kde  $\hat{S}_l[k]$  je odhad spektra řečového signálu a  $\bar{N}_l[k]$  je odhad spektra šumu získaného průměrováním spekter segmentů s pauzou v řeči. K nalezení částí s řečí a bez řeči slouží detektor řečové aktivity (VAD - Voice Activity Detector). Toto je ilustrováno na obrázku 2.1, který ilustruje na spektrogramu nalezení segmentů obsahující pouze šum.



**Obrázek 2.1:** Ilustrace VAD na spektrogramu. Šrafovane jsou vyznačeny segmenty obsahující pouze šum

Metoda spektrálního odečítání pracuje pouze s amplitudovými spektry. Fázová spektra jsou uložena a použita při rekonstrukci do časové oblasti. Průběh celého procesu zobrazuje následující schéma.

**Obrázek 2.2:** Schéma metody spektrálního odečítání

Pro praktickou implementaci slouží několik rodin algoritmů. Tato metoda má přes svojí dobrou účinnost i některá negativa. Jak je patrné z rovnice (2.6), efektivita této metody závisí na správném odhadnutí spektra šumu, což je velice obtížné ve většině případů.

## 2.4 Zvýrazňování řeči za pomoci neuronových sítí

Neuronová síť je výpočetní model používaný v umělé inteligenci. Jedná se o strukturu, která slouží pro paralelní zpracování dat. Vzorem pro její vytvoření bylo chování biologických neuronových struktur. Nacházejí uplatnění v široké škále aplikací a v širokém spektru oborů. Využívají se např. pro segmentaci, klasifikaci či kompresi dat, v akustických aplikacích pro rozpoznávání řeči či řečníka [2], potlačování šumů v signálech [3], či rozpoznání příznaků Parkinsonovy choroby, pro predikci vývoje časových řad a mnoho dalšího.

Neuronové sítě se ukázaly jakožto dobrý nástroj při zvýrazňování řeči v signálech díky jejich nelinearitě a schopnosti mapování komplexního a velmi složitého vztahu mezi zašuměnými řečovými signály a jejich čistými ekvivalenty. Další výhodou neuronových sítí je jejich schopnost potlačit různé druhy šumů. Zde pak velice záleží i na trénovacím datasetu, jelikož neuronová síť si neumí poradit s jiným typem šumu, než jaký byla trénována. Proto je potřeba zajistit co nejvíce heterogenní šumový dataset [4]. Tyto neuronové sítě nachází využití pro potlačování šumů v řečových signálech pro poslech, či jako součást videokonferenčních aplikací. Taktéž slouží jako preprocessingová část pro nástroje rozpoznání řeči, řečníka, či převodu řeči na text.

První neuronové sítě pro tento účel se objevily koncem 20. století. Jednalo se o malé neuronové sítě (SNN - Shallow Neural Network), které sloužily buď přímo k mapování zašuměného signálu na čistý [5], nebo jako měřič SNR pro různá frekvenční pásma [4]. V práci [5] je využita malá neuronová síť o hloubce čtyř vrstev, v každé z nich je 60 neuronů. Tato neuronová síť sloužila k mapování 60 vzorků zašuměného signálu na signál čistý. Problémem těchto sítí při učení bylo jejich zaseknutí se v lokálním minimu, přičemž s rostoucí hloubkou sítě, tento problém narůstal. Tento problém byl vyřešen tzv. předtrénováním, kdy je každá vrstva předtrénovaná nesupervizovaně na vstupní data a až následně následuje supervizované trénování [6]. Dnešní nástroje pro tvorbu neuronových sítí již disponují lepšími optimalizačními algoritmy, tudíž toto předtrénování není v dnešní době nutné. V dnešní době převažují neuronové sítě, které nepracují se signálem v časové oblasti, ale ve frekvenční tj. se spektry, konkrétně s jednou polovinou symetrického spektra v rozsahu  $0 \div f_s/2$  [7], [3], [8].

### 2.4.1 Neuronové sítě

Neuronová síť je složena z neuronů, které jsou mezi sebou propojeny a navzájem si předávají signály, které transformují pomocí aktivačních funkcí. Každý neuron má libovolný počet vstupů a právě jeden výstup, který může sloužit jako vstup pro více neuronů. Každý ze vstupů neuronu  $X_0 \dots X_m$  má přiřazenou váhu  $w_0 \dots w_m$ . Vstup  $x_0$

není obvykle spojen s výstupem jiného neuronu, ale má přiřazenou číselnou hodnotu a slouží jako bias  $b$ . Výstup neuronu je pak vypočten jako:

$$Y = \varphi\left(\sum_1^m X_m w_m + b\right), \quad (2.7)$$

kde  $\varphi$  je tzv. aktivační funkce, někdy též nazývána jako přenosová funkce. Jejím úkolem je zavedení nelinearity do výpočtu, jelikož násobení konstantou a sčítání jsou lineární operace. Nelinearita je v neuronových sítích klíčovým faktorem, jelikož není možné aproximovat složité vztahy mezi vstupem a výstupem sítě pouze za pomoci lineárních vztahů, a to i v případě vysokého počtu neuronů. Nejrozšířenějšími aktivačními funkcemi jsou:

1. ReLu: V současné době se jedná o nejrozšířenější aktivační funkci. Jedná se o po částech lineární funkci jejímž výstupem je vstupní hodnota pro kladná čísla a nula pro nulu a záporná čísla. Matematicky je možné tuto funkci vyjádřit jako:

$$f(x) = \max(0, x) \quad (2.8)$$

Mezi její výhody patří její výpočetní jednoduchost, lineární chování, které zjednodušuje optimalizaci sítě a dobrý zpětný přenos chyby při procesu učení viz kap. 2.4.2, [9]

2. Tanh: Jedná se o aktivační funkci esového tvaru, která převádí vstupní hodnoty do výstupního intervalu  $[-1,1]$ . Matematicky je možné tuto funkci vyjádřit jako:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (2.9)$$

Její výhodou oproti ReLu je možnost zpracovávat i záporná čísla na vstupu. Další její výhodou je její spojitost, která zabraňuje skokovým změnám ve výstupních hodnotách. Její nevýhodou je horší zpětný přenos chyby při procesu zpětné propagace chyby.

3. Sigmoid: Jedná se také o aktivační funkci esovitého tvaru. Převádí vstupní hodnoty do výstupního intervalu  $[0,1]$ . Matematický zápis této funkce je:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

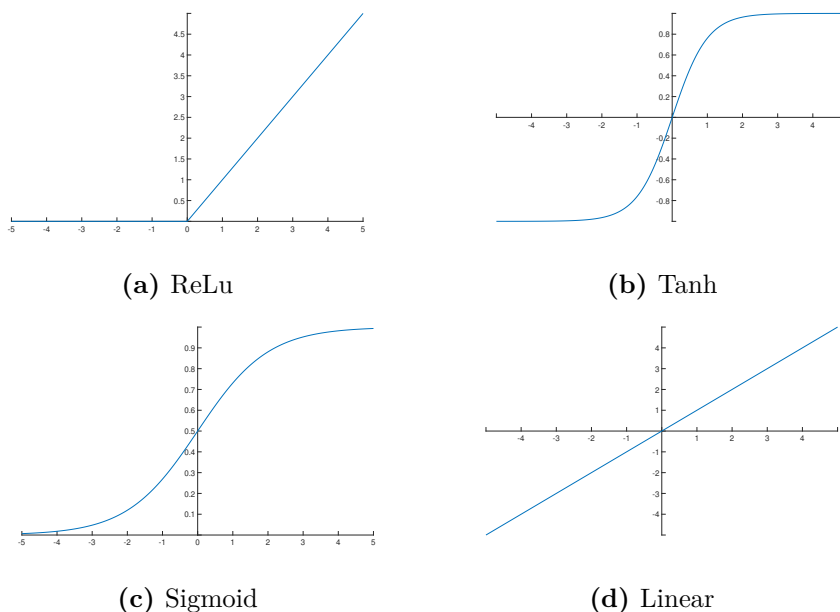
Její nevýhodou je rozsah výstupních hodnot pouze v kladných číslech. Podobně jako Tanh má horší zpětný přenos chyby.

4. Linear: Lineární aktivační funkce nalézají využití pouze ve výstupní vrstvě, která slouží výhradně pro transformaci na tensor jehož dimenze odpovídají dimenzi výstupního tensoru. Ve skrytých vrstvách se nepoužívá, jelikož nepřináší žádnou

nelinearitu do výpočtu. Při použití ve skrytých vrstvách, dochází k degradaci vícevrstvé neuronové sítě na síť jednovrstvou, jelikož poslední vrstva by byla stále lineární funkcí vstupu sítě.

Existují však i další aktivační funkce, přičemž některé z nich jsou odvozené od těchto např. ELU, LeakyRelu a další.

Obrázek 2.3 a-d zobrazují základní aktivační funkce.



**Obrázek 2.3:** Čtyři základní aktivační funkce

Neurony se skládají do vrstev, přičemž neurony v jedné vrstvě nejsou mezi sebou propojeny. Tyto vrstvy se skládají za sebe a neurony jsou propojeny právě mezi sousedními vrstvami. Existuje několik možností propojení. Nejčastější z nich je plné propojení, kde výstup každého neuronu  $n$ -té vrstvy je připojen na vstup každého v  $n + 1$  vrstvě.

Základní neuronová síť standardně obsahuje 3 druhy vrstev: vstupní, skryté, výstupní. Vstupní vrstva slouží pouze jako vstup neuronové sítě a nemá aktivační funkci. Jejím účelem je pouze propojení mezi vstupními daty a skrytými vrstvami (nejčastěji s první skrytou). Skryté vrstvy jsou hlavní vrstvy, na kterých je prováděn výpočet. Počet neuronů v nich obsažených není závislý na formátu vstupních, či výstupních dat. Výstupní vrstva slouží jako transformace mezi poslední skrytou vrstvou a výstupními daty. Obsahuje také aktivační funkci. Počet neuronů této sítě je závislý na formátu výstupních dat, kdy počet neuronů odpovídá počtu prvků výstupního tensoru.

## 2.4.2 Učení neuronové sítě

Existují dva základní způsoby učení neuronových sítí a to s učitelem (supervised) a bez učitele (unsupervised). První z nich vyžaduje dva sety dat: a to data originální a data referenční (samples a labels) tj. pro každé ze vstupních dat existuje referenční. Při procesu učení se síť učí mapovat vstupní data na data referenční. Při inicializaci neuronové sítě jsou nastaveny váhy v jednotlivých neuronech. Váhy mohou být nastaveny náhodně, na konstantní hodnotu, s gausovským rozložením, či jinak. Při učení neuronové sítě jsou síti postupně předkládána originální data na vstup. Z těchto dat síť na výstupu vytvoří predikci, ta je následně porovnána s referenční hodnotou pomocí ztrátové funkce (loss function). Tento rozdíl/chyba je následně poslán zpět neuronovou sítí (backpropagation) a podle ní se upraví váhy a bias na jednotlivých neuronech. Cílem je aby se predikovaná data lišila co nejméně od dat referenčních [10].

Při učení bez učitele se algoritmus učí bez nutnosti referenčních dat (labels). Jedním z nejčastějších příkladů učení bez učitele je autoencoder, což je síť, která je učena rekonstruovat svůj vlastní vstup. Nejčastěji se využívá pro komprese dat [11].

### Důležité pojmy:

Rychlost učení (LR -learning rate) - Je jedním z hyperparametrů, který určuje, jak moc se změní váhy a biasy neuronů v reakci na chybu predikce. Běžná hodnota tohoto parametru je mezi 0.0 a 1.0. LR určuje jak rychle se model adaptuje na řešený problém. Menší LR vyžaduje větší množství epoch k natrénování v důsledku menších změn. Větší hodnota LR má za následek rychlejší učení, existuje zde však riziko příliš velké změny tj. že váhy a biasy se změní příliš a dojde k překročení optimálních hodnot a tudíž nakonec síť nevykonverguje k optimálnímu řešení. Tento parametr je možné v průběhu učení měnit. V praxi se toho často využívá. V počátku učení (v dřívějších epochách), kdy se predikované hodnoty od referenčních velmi liší, se nastaví vyšší LR pro rychlejší učení a v pozdějších fázích (epochách), kdy je třeba jen menších změn se snižuje, aby síť konvergovala ke globálnímu minimu chyby a tudíž k nejpřesnějším predikcím. Optimální hodnotu LR nelze obecně stanovit. Vždy záleží na konkrétní struktuře sítě a datech [12].

Epocha - Epochou se rozumí průchod celého datasetu neuronovou sítí. Při učení sítě se využívá více epoch. Počet epoch pro naučení neuronové sítě závisí na struktuře sítě, datech i nastavení dalších hyperparametrů. V některých případech postačí 10 epoch v jiných je třeba i několik set epoch.

Velikost dávky (Batch Size) - Při učení neuronové sítě jsou standardně data rozdělena

do tzv. dávek/bloků (batch). Batch-size určuje počet prvků v každé dávce. Ke zpětné propagaci a změně vah a biasů dochází po průchodu každého bloku. Chyba je stanovována z celého bloku ne jen z jednoho vzorku dat. v případě že velikost počet vzorků v datasetu není dělitelný velikostí dávky, je poslední dávka epochy menší. Batch size je velmi důležitým parametrem ovlivňující výsledky neuronové sítě. Větší dávka má za následek větší generalizaci, což je žádoucí. Příliš velká generalizace je již ovšem nežádoucí, jelikož síť pak není schopna se naučit jednotlivé detaily na jednotlivých vzorcích a učí se jen převažující trendy. Další nevýhodou větších velikostí může být problém s nedostatkem operační paměti pro uchování celého bloku. Výhoda menšího bloku spočívá v tom, že nepřekročí kapacitu operační paměti. Další výhodou je vyšší stabilita a menší chyba způsobená generalizací. Příliš malý batch size má však za následek tzv. overfitting tj. že neuronová síť dokáže velmi dobře zpracovat data na kterých byla učena ale nedokáže generalizovat. Standardně se volí batch size v rozmezí jednotek až nižších stovek [13].

Ztrátová funkce (Loss Function) - je funkce, která při učení sítě porovnává predikované hodnoty s hodnotami referenčními. Ukazuje jak dobře je neuronová síť natrénovaná. Při procesu učení je cíleno na minimalizaci hodnot této funkce. Existuje mnoho ztrátových funkcí, jejich použití záleží na účelu a tudíž i na struktuře dané sítě. V základu je možné tyto funkce rozdělit na dvě kategorie na regresní a klasifikační. Klasifikační se využívají v klasifikačních neuronových sítích, kdy ze vstupu, jímž mohou být nejrůznější data síť vytvoří na výstupu vektor pravděpodobností, kde každý prvek vektoru odpovídá předpřipravené kategorii. Nejčastěji používané klasifikační funkce jsou Binary Cross Entropy a Categorical Cross Entropy. Regresní se používají v regresních neuronových sítích, kdy pro daný vstup síť vytvoří odpovídající výstup. Nejčastěji používané regresní funkce jsou MSE (Mean Squared Error) a MAE (Mean Absolute Error) [10].

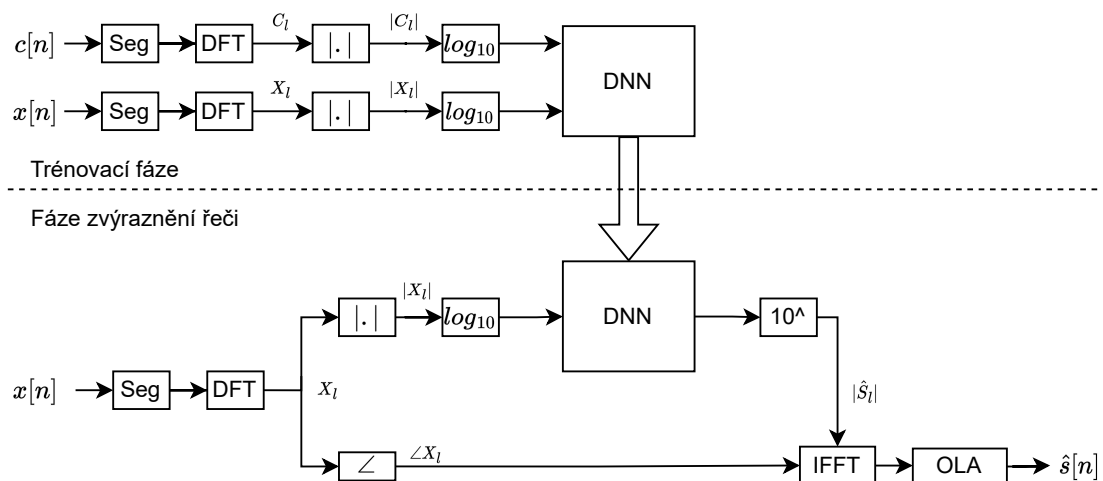
### 2.4.3 Optimalizační algoritmy

Jak již bylo zmíněno v kapitole 2.4.2, při procesu učení dochází ke zpětné propagaci chyby a podle ní se mění váhy a biasy. Za to, jak se změní, odpovídají právě optimalizační algoritmy. Každý z algoritmů má své výhody a nevýhody a hodí se pro jiné aplikace. Nejčastěji používané algoritmy jsou Gradient descent, Stochastic gradient descent, RMS PROP a Adam [14].



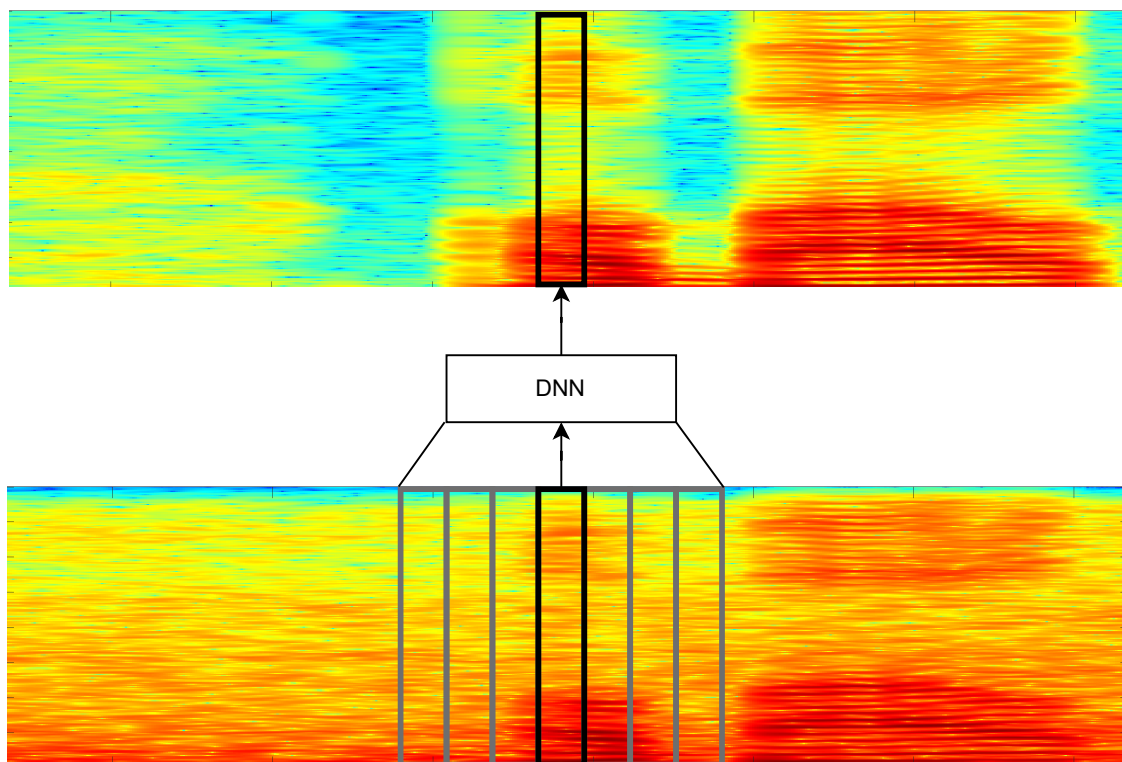
## 2.5 Neuronová síť na bázi spektrálního odečítání

Struktura použitá v tomto projektu vychází ze struktury spektrálního odečítání. V této struktuře je nahrazena část metody spektrálního odečítání, konkrétně odhad spekter šumu a jeho následné odečítání, neuronovou sítí. Rozdíl mezi oběma přístupy je patrný z orázků 2.2 a 2.4.



Obrázek 2.4: Schéma metody využité v práci

Při využití neuronové sítě je postup rozdělený do dvou částí. V první části - trénovací fázi se neuronová síť učí mapovat amplitudová spektra zašuměného signálu na spektra čistého referenčního signálu. Na vstup neuronové sítě jsou postupně přiváděna spektra zašuměného signálu a jejich spektrální okolí a na výstup jejich ekvivalenty bez šumu, bez okolí viz obrázek 2.5. Využití okolí na vstupu má výhody při odstraňování konvolučních šumů a zároveň pomáhá se zachováním harmonické struktury řeči.



**Obrázek 2.5:** Ilustrace zvýraznění řeči pomocí DNN

Ve druhé části, tj. ve fázi zvýraznění řeči je již natrénovaná neuronová síť využita pro úpravu amplitudového spektra, podobně jako odečtení spekter šumu v Metodě spektrálního odečítání, ale jako vstup stále slouží centrální spektrum + okolí. Pro rekonstrukci výstupních segmentů signálu  $\hat{s}_l$  slouží opět fáze vstupu  $\angle X_l$ .

Zadané podmínky pro tuto metodu byly: segmentace s 50% překryvem s využitím Hammingova váhovacího okna, vzorkovací frekvence  $f_s = 8000$  Hz, počet vzorků v segmentu 256.

## 3 Implementace

Pro přípravu dat byl využit programovací jazyk Python a jeho následující knihovny:

1. Numpy: Knihovna umožňující práci s vícedimensionálními objekty (vektory, matice atd.). Umožňuje matematické operace, logické operace, seřazení, vyhledávání, Fourierovy transformace, manipulace s tvarem, základní operace lin. algebry, statistické operace a další [15].
2. SciPy: Rozšíření knihovny NumPy. Její součásti jsou více zaměřeny na statistiku a zpracování signálů [16].
3. Librosa: Knihovna pro hudební a audio analýzu. V projektu slouží především v preprocessingové části pro načítání a ukládání signálů a jejich převzorkování [17].
4. Soundfile Slouží pro načítání, ukládání a další manipulaci s zvukovými soubory [18].
5. Cmath Knihovna umožňující práci s komplexními čísly [19].

Pro samotné učení neuronové sítě byl taktéž využit Python a knihovny TensorFlow [20] a Keras [21]. Tensorflow je nástroj pro strojové učení a numerické simulace. Nabízí High-level interface pro vytváření i trénování modelů strojového učení, stejně tak nabízí i low-level interface pro pokročilé uživatele. Využívá dataflow graph model pro reprezentaci výpočtu (učení), kde vrcholy v grafu reprezentují matematické operace a hrany grafu tok informací mezi nimi. Tensorflow nabízí podporu mnoha platforem a zařízení. Výpočty mohou být prováděny na CPU, GPU, TPU a to i paralelně. Další výhodou je možnost vytvoření vstupních streamů (pipeline) a transformací, což má výhodu především u větších datasetů, které jsou větší než dostupná operační paměť přístroje, kde k výpočtu dochází.

Keras je vytvořen jako nadstavba pro Tensorflow pro vytváření a učení neuronových sítí. Výhodou tohoto nástroje je rozhraní, které umožňuje vývojáři/výzkumníkovi vytvořit jednoduše i poměrně komplikované neuronové sítě. Podporuje širokou škálu vrstev: konvoluční, rekurentní, plně i prppojené. Navíc také umožňuje vytvářet vlastní vrstvy, ztrátové funkce aj.

Všechny výpočty byly prováděny s využitím MetaCentra [22], což je česká gridová infrastruktura patřící pod CESNET, z. s. p. o. jehož součástí je i několik univerzit a některá pracoviště Akademie věd ČR [23]. MetaCentrum umožňuje členům akademické obce provádět výpočty a simulace na sdílených výpočetních clusterech.

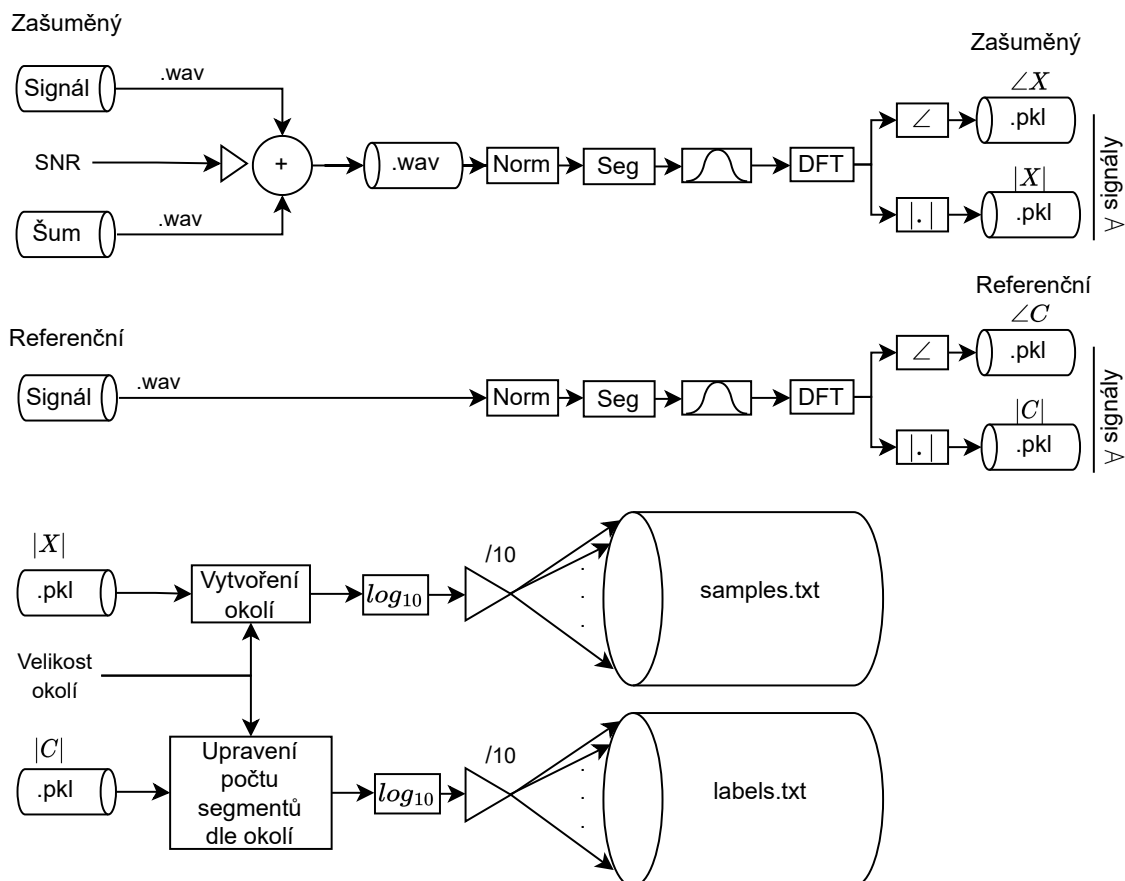
## 3.1 Implementace přípravné části

Přípraná část byla oddělena od samotného učení neuronové sítě z důvodu úspory času při učení, jelikož bylo na stejných datech trénováno více modelů. Přípravná část se skládá ze tří základních částí.

1. Zašumění signálu: V první fázi je jsou smíchány signály s šumy při požadovaných SNR a výsledné signály uloženy.
2. Vytvoření spekter: V této fázi dochází k segmentaci (overlap 50%, hammingovo okno [24]) a vytvoření spekter pomocí FFT pro tyto segmenty. Spektra jednotlivých souborů jsou následně uložena. Tento krok zahrnuje čisté signály i signály zašumněné.
3. Vytvoření finálního souboru: V poslední fázi jsou načtena spektra, mohou být převedena na logaritmická a dále normalizovaná viz kapitola 4.3.2. U zašuměných signálů jsou ke spektrům přidána ještě okolí. Následně jsou všechna zašuměná spektra uložena do jednoho souboru a čistá do jednoho. Toto má za následek zrychlení trénování neuronové sítě, jelikož není nutné otevírat více souborů, což stojí výpočetní čas. Jako formát pro uložení dat byl vybrán \*.txt, kvůli rychlosti načítání [25]. Další z možností uložení byl \*.csv soubor, jeho využití by proces ukládání, či načítání neurychlil [26].

Implementace je graficky znázorněna na obrázku 3.1. Přípravné části slouží dvě autorské knihovny a to `signal_op.py` a `file_preps.py`. Obě dvě knihovny využívají knihovny zmíněné výše. Funkce také využívají funkce z dvou zmíněných knihoven.

*Pozn. Pro stručnost v následujících seznamech nejsou uvedeny funkce, které dané funkce využívají pro řešení parciálních úkolů. Dále v textu nebudou zmíněny funkce, které slouží jen pro vytvoření adresářů a volání dalších funkcí, které jsou zodpovědné za obsah. Vybrané funkce slouží jako ilustrace.*



Obrázek 3.1: Schéma metody využité v práci

První zmíněná knihovna se zabývá samotným zpracováním signálu načteným do pythonovského listu, či do `numpy.ndarray`. Obsahuje následující nejdůležitější funkce:

1. `mix_at_given_SNR` - Slouží ke smíchání signálu a šumu při daném SNR.
2. `get_random_part_of_noise` - Vybere náhodný úsek požadované délky z šumového signálu.
3. `make_spectrum_of_signal_frame` - Slouží k vytvoření komplexního spektra, amplitudového a fázového. Komplexní i amplitudová spektra jsou pouze jednostranná tj. jsou zde zastoupeny frekvence v rozmezí  $0 \div f_s/2$ , kde  $f_s$  je vzorkovací frekvence signálu.
4. `mv_normalize` - Normalizuje vstupní data na nulovou střední hodnotu a rozptýl roven jedné.
5. `abs_normalize` - Normalizuje vstupní data tak, že jejich maximální absolutní hodnota je rovna jedné.

6. `to_log_spec` - Převádí spektra na logaritmická. Pro omezení výstupního rozsahu jsou vstupní data menší než  $1e-9$  nastavena na hodnotu  $1e-9$ .
7. `to_real_spec` - Převádí logaritmická spektra na amplitudová nelogaritmická.
8. `make_segments_from_signal` - Vytvoří segmenty požadované délky s požadovaným překryvem s využitím vyžádaného váhovacího okna.
9. `make_complex_number` - Ze zadané amplitudy a fáze vytvoří komplexní číslo.
10. `reconstruct_signal` - Z listů, které obsahují amplitudová a fázová spektra vytvoří se znalostí překryvu signál v časové oblasti.
11. `make_spektrum_areas` - Z listu obsahující amplitudová spektra vytvoří list kde každá položka odpovídá sloučení  $n$  spekter předcházejících, aktuálního spektra a  $n$  spekter následujících tj. přidá na obě strany centrálního spektra kontext-okolí.
12. `denoise_signal` - S využitím natrénovaného modelu provede redukci šumu v řečovém signálu.
13. `measure_snr` - S využitím čisté alternativy zašuměného signálu vypočte SNR.
14. `measure_pesq` - S využitím čisté alternativy signálu vypočte PESQ.

Druhá z nich se zabývá prací se soubory. Obsahuje tyto nejdůležitější funkce:

1. `mix_speeches_with_noises_files` - Smíchá řečové signály ze složky s šumy ze složky při daném SNR/daných SNR. Šum k dané promluvě je vybírán náhodně s rovnoměrným rozložením pravděpodobnosti.
2. `mix_speeches_with_noises_files_SNR_folders` - Jedná se o obdobu předchozí funkce. Tato funkce ale míchá signály s šumy při pevně daných hodnotách SNR a každý ze signálů je s daným šumem smíchán při všech hodnotách SNR. Každé SNR má svoji výstupní složku.
3. `load_signal` - Načte signál z `*.wav` souboru a převzorkuje ho na zadané  $f_s$ .
4. `save_signal` - Uloží signál do `*.wav` souboru.
5. `create_spectrum_file_from_signal_file` - Načte signály ze souborů, provede segmentaci, převedení segmentů do spekter. Amplitudová a fázová spektra jsou uložena po vstupních souborech do `*.pkl` souborů.

6. `create_spectrum_file_from_signal_file_folders` - Funkce obdobná předchozí. Oproti ní však pracuje se signály uloženými do více složek podle jejich SNR (soubory vytvořené funkcí `mix_speeches_with_noises_files_SNR_folders`)
7. `load_spectrums_from_folders` - Načítá \*.pkl soubory obsahující spektra. Výstupem je list obsahující list spekter z každého souboru.
8. `load_spectrums_from_folders_and_make_areas` - Funkce obdobná předchozí. Oproti ní však vrací list listů spekter s přidaným okolím.
9. `load_spectrums_from_folder_and_make_text_file_real` - S využitím předchozích 2 funkcí načte spektra ze složek. V případě že se jedná o samples načte je s požadovaným okolím, v případě labels bez. Výstupem této funkce je \*.txt soubor obsahující na každém řádku jedno spektrum resp. jedno spektrum s okolím. K této funkci existuje také alternativa pro vícesložkové načítání, zde je však nutné vzít v úvahu, že složek obsahující zašuměná data je více, ale čistá referenční data mají pouze složku jednu a je nutné je tedy do \*.txt souboru zapsat vícekrát.

Pro funkce směřující řeč se šumem, vytvářející spektrální soubory a finální textovou reprezentaci databáze existují i verze podporující vícevláknové/víceprocesové zpracování.

## 3.2 Implementace vstupního streamu (pipeline)

Pro načtení předpřipraveného datasetu v podobě \*.txt souboru jsou možné dva přístupy. Jedním z nich je standardní načtení do operační paměti, druhým z nich je vytvořit vstupního streamu. Problémem načtení datasetu do operační paměti je potenciální nedostatek místa v této paměti. Toto je však možné vyřešit postupným načítáním po menších dávkách (batch). Výhodou je možnost pracovat přímo s načtenými daty při jejich předzpracování pro neuronovou síť.

Při vytvoření vstupního streamu nejsou data načtena do operační paměti. Data jsou postupně načítána až při samotném trénování neuronové sítě, tudíž nehrozí nedostatek místa v operační paměti. Dalším rozdílem je, že kód není psán pro přímou manipulaci s daty, ale jako sadu instrukcí, která má být aplikována, až budou data např. z jednoho řádku načtena. Největší výhodou tohoto řešení je paralelizace procesů, jelikož data nejsou načítána před zahájením učení neuronové sítě, ale v průběhu. V momentě, kdy neuronová síť pracuje s  $n$ -tým vzorkem, je již načítán  $n + 1$ . Pro ještě větší urychlení je možné využít i tzv. precache tj. že nebude načten jen  $n + 1$

vzorek, ale i další. Množství načtených vzorků je limitováno velikostí RAM, nebo může být limitováno přímo v kódu. Pro vytvoření vstupního streamu (pipeline) má TensorFlow třídu `tf.data.Dataset` a její potomky, kteří umožňují načítat různé druhy datasetů např. binární soubory, textové soubory, CSV soubory a dále s nimi pracovat. Načtená data jsou reprezentovaná jakožto tenzor `tf.Tensor`. Tensorflow umí pracovat i s jinými formáty dat např. `numpy.array`, avšak tenzor je základní reprezentace dat. Pro všechny běžné datové typy existje v TensorFlow jejich tensorové vyjádření.

---

```
samples = tf.data.TextLineDataset(home+"/"+sam)
labels = tf.data.TextLineDataset(home+"/"+lab)
```

---

V prvním kroku jsou vytvořeny vstupní streamy (pipeline). Data budou načítána po řádcích, tudíž budou reprezentována jako stringový tenzor, je třeba je upravit a to konkrétně odříznou na konci řádku znak `"\n"`, rozdělit řádek na jednotlivá čísla (stále ještě reprezentovaná jako `string`) tj. z každého řádku vytvořit stringový tenzor délce odpovídající počtu hodnot na řádku a následně převést tento tenzor na tenzor s hodnotami typu `float64`.

---

```
def decode(line):
    #odstraneni \n na konci radku
    line = tf.strings.strip(line)
    #rozdeleni radku
    line_split = tf.strings.split(line, "\n")
    #prevedeni na float
    numbers = tf.strings.to_number(line_split)
    return numbers

samples = tf.data.TextLineDataset(home+"/"+sam)
    .map(decode, num_parallel_calls=tf.data.AUTOTUNE)
    .prefetch(tf.data.AUTOTUNE)
labels = tf.data.TextLineDataset(home+"/"+lab)
    .map(decode, num_parallel_calls=tf.data.AUTOTUNE)
    .prefetch(tf.data.AUTOTUNE)
```

---

V závěru jsou oba dva datasety `samples + labels` spojeny v jeden dataset.



### 3.3 Implementace neuronové sítě

Jak již bylo zmíněno výše, pro implementaci neuronové sítě byla využita knihovna Keras z nástroje TensorFlow. Kód níže ukazuje její implementaci.

---

```
model=Sequential([
    Dense(1500, activation="tanh", input_shape=(129*area,)),
    Dense(1500, activation="tanh"),
    Dense(1500, activation="tanh"),
    Dense(129, activation="linear"),
])
model.compile(optimizer=Adam(),
              loss = 'mean_squared_error',
              metrics=['mean_squared_error', 'accuracy'])

model.fit(dataset, epochs = epochs,
          verbose = 2
          callbacks=[tensorboard_callback,
                    tf.keras.callbacks.LearningRateScheduler(decay)])
```

---

V první části je vytvořen samotný model, jsou do něj dány vrstvy, nastaven jejich počet neuronů a aktivační funkce. V další části dochází ke kompilaci modelu a nastavením ztrátové funkce a optimalizačního algoritmu. V poslední části `model.fit` spustí učení neuronové sítě na zadaném datasetu. Do parametrů této funkce jsou zařazeny dva callbacky: `tensorboard_callback` slouží pro ukládání dat z učení, která je pak možno vizualizovat. `LearningRateScheduler(decay)` zajišťuje změnu rychlosti učení v průběhu epoch, tím že volá funkci `decay`, která určuje rychlost učení v závislosti na epoše.

---

```
def decay(epoch):  
    if epoch < 10:  
        return 0.0003  
    elif epoch >= 10 and epoch < 20:  
        return 0.00008  
    else :  
        return 0.000008
```

---

## 4 Experimenty a výsledky

Tato kapitola popisuje experimentální část projektu a její výsledky. Experimenty probíhaly na několika sítích a testovala se jejich jednotlivá nastavení a také formát vstupních dat. Cílem experimentu bylo analyzovat funkčnost dopředných neuronových sítí pro zvýrazňování řeči tj. jaké úrovně potlačení šumu lze dosáhnout a jaké bude zkreslení výstupního signálu. Pro kvantifikaci potlačení šumu a zkreslení byla použita dříve popsaná kritéria WADA SNR a PESQ.

Pro experimenty byla stěžejní data. Řečové signály byly získány z několika zdrojů. První dataset, dále v textu nazývaný T-Dataset, byl poskytnut pracovištěm vedoucího práce. Obsahuje 2445 promluv o průměrné délce 6,3 s. Dataset je složen z promluv 18 řečníků a jsou zde zastoupeni muži i ženy. Jednotlivé promluvy obsahují taktéž ticho před a za promluvou.

Šumové data byla získána z několika zdrojů. Prvním z nich je dataset poskytnutý pracovištěm vedoucího práce, který obsahuje několik hodin dynamických šumů, a to konkrétně nahrávek zvuků z prostředí: restaurace, kuchyně, silnice, z vnitřku auta za jízdy. Dalším datasetem je MS-SNSD Dataset [27], který obsahuje 128 nahrávek zvuků běžného dne (Mluvení v pozadí, dveře, kopírka, psaní na klávesnici a další)

Příprava databází pro učení a ověřování funkčnosti neuronových sítí byla oddělena od samotného učení, jelikož bylo zkušeno více neuronových sítí a jejich nastavení na jedné testovací databázi. Příprava byla rozdělena do tří částí.

1. Zašumění signálu: V první fázi jsou smíchány signály s šumy při požadovaných SNR a výsledné signály uloženy.
2. Vytvoření spekter: V této fázi dochází k segmentaci (overlap 50%, hammingovo okno [24]) a vytvoření spekter pomocí FFT pro tyto segmenty. Spektra jednotlivých souborů jsou následně uložena. Tento krok zahrnuje čisté signály i signály zašuměné.
3. Vytvoření finálního souboru: V poslední fázi jsou načtena spektra, mohou být převedena na logaritmická a dále normalizovaná viz kapitola 4.3.2. U zašuměných signálů jsou ke spektrům přidána ještě okolí. Následně jsou všechna zašuměná spektra uložena do jednoho souboru a čistá do jednoho. Toto má za následek zrychlení trénování neuronové sítě, jelikož není nutné otevírat více souborů, což stojí výpočetní čas. Jako formát pro uložení dat byl vybrán \*.txt, kvůli rychlosti načítání [25]. Další z možností uložení byl \*.csv soubor, jeho využití by proces ukládání, či načítání neurychlil [26].

Jelikož chování neuronové sítě a její výsledky ovlivňuje veliké množství faktorů, bylo nutné otestovat veliké množství různých kombinací parametrů. Kvalita sítě byla kvalifikována za použití testovacího datasetu, který obsahoval 100 čistých řečových signálů a jeho zašuměné ekvivalenty při -5 dB, 0 dB, 5 dB, 10 dB, 15 dB, 20 dB. Ke každému z datasetů určených k trénování sítě, existoval jeden takovýto testovací. Data z testovacího datasetu nebyla využita při trénování DNN. Parametry, které se sledovaly po zvýraznění řeči v těchto zašuměných signálech byly WADA SNR a PESQ. Dalším kritériem bylo subjektivní posouzení kvality poslechu. Posledním kritériem byla míra přesnosti predikce, tj. míra podobnosti mezi predikovanými a referenčními spektry při procesu učení tzv. accuracy (Acc).

## 4.1 Databáze využité pro trénování

Jak bylo zmíněno výše v kapitole 4 z databází řeči a šumů bylo nutné vytvořit finální databáze promluv smíchaných s šumy a jejich čisté reference. Tato sekce se zabývá finálním složením jednotlivých trénovacích datasetů. Finální datasety (logaritmická spektra) byla před trénováním DNN přeškálována, viz kapitola 4.3.2. Přeškálování zajistilo distribuci hodnot spektrálních čar do intervalu  $[-1, 1]$ , což je výhodné z několika důvodů. Prvním z nich je předcházení explodujícího gradientu, tj. předcházení vysokých hodnot gradientu, které vedou k divergenci sítě. Druhým z nich je výhodnost rozsahu těchto hodnot pro tanh aktivační funkci, která byla v experimentech využita.

První z databází byla **T1500+CAR**. Byla vytvořena smícháním 1551 promluv z T datasetu a 35 šumových signálů, komplexních zvuků dopravy, rušných ulic a vozidel. Signály a šumy byly smíchány mezi 8 a 20dB.

Další z databází byla **T1500+GEN\_NOISES**. Byla vytvořena smícháním 1551 promluv z T datasetu a 125 zvuků obsahujících např. Mluvení, zvuk kopírky, hlášení na letišti, vysavač, skřípající židli, zavírající se dveře, vysavač atd. K tomuto navíc byla přidána data z předchozího datasetu.

## 4.2 Prvotní experimenty

Tato sekce přibližuje první pokusy, z jejichž výsledků a z literatury vycházely pozdější experimenty. Jelikož se dobrá nastavení z těchto experimentů objevují v pozdějších experimentech a ta špatná nevedla k dobrým výsledkům, budou zde výsledky pouze diskutovány, bez uvedení dat.

První experimenty sloužily k seznámení se s problematikou a ustavení ideální DNN pro další experimenty. V této fázi byly také vyzkoušeny různé struktury plně propojených dopředných DNN a různé aktivační funkce.

Z aktivačních funkcí se ukázala jako nejlepší pro daný úkol a daná data tanh funkce, jelikož umí pracovat i s kladnými i se zápornými hodnotami. Její nevýhoda spočívá v její klesající citlivosti na změny kolem 1 a -1, ale tento problém je možné vhodným škálováním dat odstranit. Funkce sigmoid a ReLu se ukázaly jako nevhodné, jelikož část logaritmického spektra, má zápornou hodnotu se kterou tyto funkce neumí pracovat a změni ji na 0. Dalším potenciálním kandidátem byla funkce LeakyRelu, která se v kladných hodnotách chová jako ReLu a pro záporné hodnoty je její výstup roven  $\alpha$  krát vstupu, kde  $\alpha$  je parametr, který se nastavuje před učením neuronové sítě. Pro aplikaci v této práci se však LeakyRelu i ReLu ukázaly jako nevhodné, jelikož transformace, kterou umožňují je příliš jednoduchá.

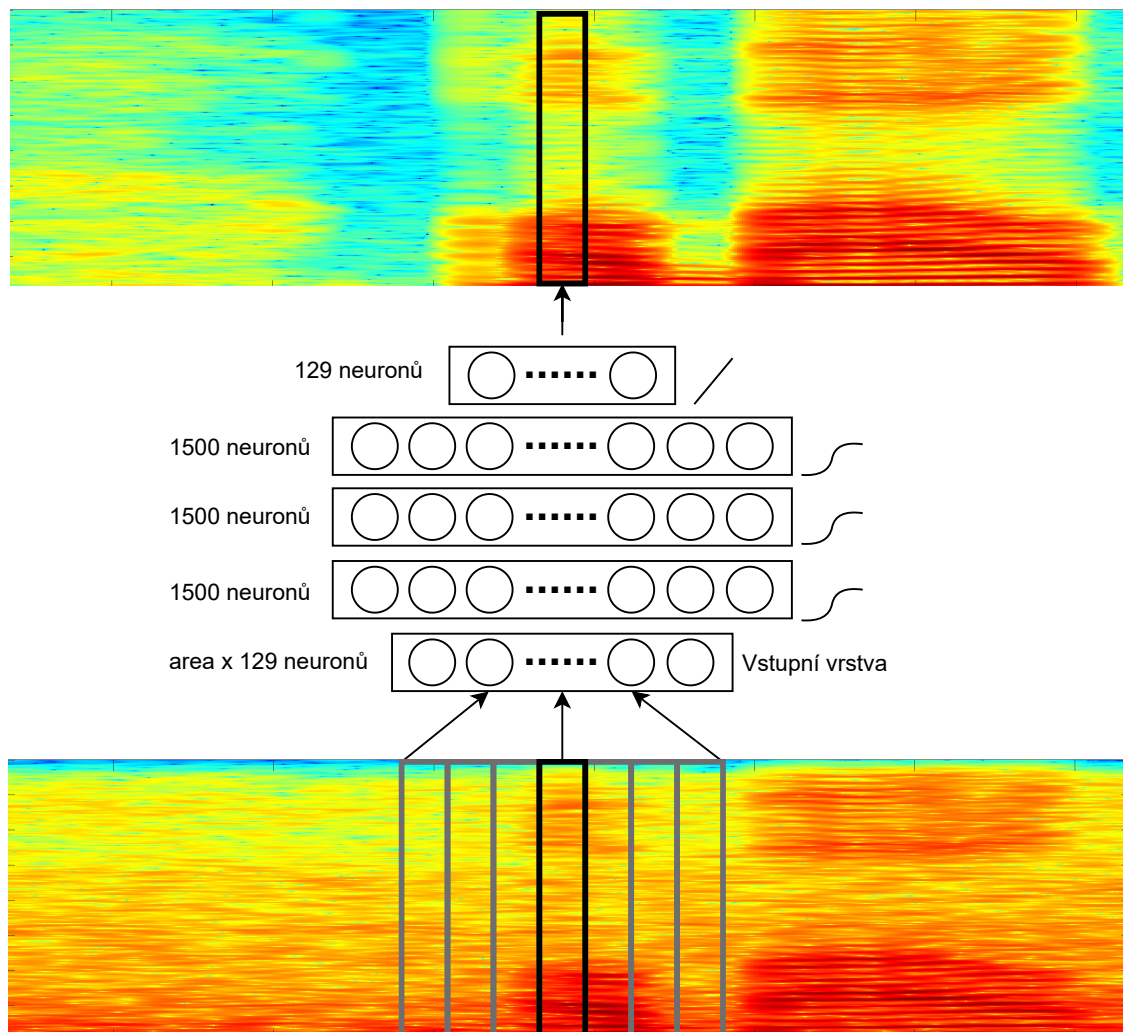
Nejčastěji používanou aktivační funkcí pro tento typ je MSE, která byla použita i v této práci. Byly vyzkoušeny také jiné ztrátové funkce jako: Střední absolutní chyba (MAE - Mean Absolute Error), či střední absolutní procentní chyba (MAPE - Mean Absolute Percentage Error). Ze všech testovaných funkcí byla nejlepší MSE.

Z hlediska optimalizačních učících algoritmů byl porovnáván algoritmus SGD, RMS PROP a Adam. Díky lepším výsledkům, byl pro práci zvolen Adam.

## 4.3 Experiment 1: 3 skryté vrstvy po 1500 neuronech a T1500+CAR

Tato neuronová síť se skládá ze vstupní vrstvy, třech skrytých vrstev po 1500 neuronech a tanh aktivační funkcí a výstupní vrstvy o 129 neuronech a lineární aktivační funkcí. Tato poslední vrstva slouží pouze jako transformace mezi skrytými vrstvami a požadovaným výstupem, kterým je spektrum od 0 do  $f_s/2$ . Obrázek 4.1 ilustruje strukturu a fungování sítě. Jako učební dataset byl zvolen dataset T1500+CAR. Jako ztrátová funkce byla zvolena MSE a jako optimalizační algoritmus Adam. Následující podkapitoly se zabývají výsledky jednotlivých nastavení

a kombinací parametrů tohoto modelu a shrnutím vlivu nastavení jednotlivých parametrů na kvalitu výstupu sítě.



**Obrázek 4.1:** Ilustrace mapování zašuměných spekter na čistá. Parametr area odpovídá počtu spekter (centrální + okolí), která vstupují do sítě

Na této struktuře modelu bylo provedeno přes 50 experimentů, jejichž cílem bylo zjistit nejlepší kombinaci parametrů a nastavení pro nejlepší výsledek. Dále byl otestován různý formát vstupních dat a jejich normalizace. Pro učení jednotlivých modelů byla využita proměnná rychlost učení (schodovitá), která byla pro každý z modelů individuálně upravena, aby bylo možné dosáhnout co nejlepších výsledků. Model byl trénován při vstupních SNR v rozmezí 8 dB ÷ 20 dB.

### 4.3.1 Korelace SNR a WADA SNR

Před samotným testováním výsledků modelů, bylo třeba otestovat jestli výsledky dané WADA SNR algoritmem odpovídají tradičnímu SNR. Toto bylo otestováno na testovacím datasetu, čímž se vyloučil vliv zkreslení signálu na výstupu DNN. Tabulka 4.1 ukazuje srovnání SNR a WADA SNR pro testovací dataset.

**Tabulka 4.1:** Porovnání SNR a WADA SNR

SNR [dB]	-5	0	5	10	15	20
WADA SNR[dB]	-0.92	0.59	4.28	8.53	12.92	17.18

Jak je patrné z tabulky WADA SNR velmi dobře koresponduje se SNR s jakým byly signály v testovacím datasetu vytvořeny. Jediný podstatnější rozdíl se objevuje u záporného SNR, což je očekávatelné, jelikož v těchto signálech má šum větší výkon než řeč a tudíž aproximace není tak přesná.

### 4.3.2 Formát vstupních dat

První z testovaných parametrů byl formát vstupních dat. Pro aktivační funkci tanh je nejlepší, aby data byla z intervalu  $[-1,1]$ , ale existuje několik přístupů, jak tohoto docílit. Z hlediska normalizace signálů v časové oblasti bylo vyzkoušeno přeškálování signálu do rozsahu  $[-1,1]$  (abs normalizace) a normalizace na nulovou střední hodnotu a jednotkový rozptyl (MVN - Mean Variance Normalisation), obojí na úrovni jednotlivých souborů (promluv). Ve frekvenční oblasti na úrovni spekter bylo vyzkoušeno přeškálování vydělením 5, vydělením 10. Také bylo vyzkoušeno spektrum výkonové škálované vydělením 10. Dále bylo vyzkoušeno škálování abs normalizací na úrovni jednotlivých spekter, to ovšem nevedlo k relevantním výsledkům a proto již nebude dále zmiňováno. MVN na úrovni spekter za pomoci průměrné střední hodnoty a jednotkového rozptylu z celého datasetu byla taktéž zavrhnuta, byť je možné se s ní v některých publikovaných experimentech setkat, jelikož spektra by pak neopovídala ideálnímu rozsahu pro aktivační funkci tanh. Pro porovnání posloužil model s velikostí dávky 128. Jak již bylo zmíněno v dřívějších kapitolách pro všechny testy byla využita spektra logarimická. Tabulky 4.2, 4.3, 4.4 zobrazují porovnání výsledků pro abs normalizaci a MVN vstupních signálů, kde spektra byla normována přeškálováním 10.

**Tabulka 4.2:** Porovnání Acc pro různé normalizace souborů

Normalizace	Acc [%]
ABS	79.20
MVN	77.78

**Tabulka 4.3:** Porovnání WADA SNR out pro různé normalizace signálů

Normalizace	SNR in [dB]	-5	0	5	10	15	20
	WADA SNR in[dB]	-0.92	0.59	4.28	8.53	12.92	17.18
ABS	WADA SNR out[dB]	17.39	24.46	31.08	34.65	35.83	35.95
MV	WADA SNR out[dB]	13.99	18.79	21.75	22.42	21.16	19.58

**Tabulka 4.4:** Porovnání PEQS out pro různé normalizace signálů

Normalizace	SNR in [dB]	-5	0	5	10	15	20
	PESQ in	1.76	2.05	2.42	2.87	3.32	3.73
ABS	PESQ out	1.85	2.24	2.72	3.20	3.61	3.90
MV	PESQ out	1.54	1.81	2.17	2.59	3.00	3.34

Z tabulek je patrné, že normování signálu na nulovou střední hodnotu a jednotkový rozptyl vykazuje horší výsledky oproti ABS normalizaci. Toto je způsobeno tím, že oproti ABS normalizaci, která signál pouze přeskálovala, a tudíž zůstal ve spektru stejný poměr mezi spektrálními čarami, MV normalizace změnila distribuci spektrálních čar. Pro některé aplikace je toto výhodné, ovšem, jak vyplývá z tabulek, pro tuto nikoliv. V dalších experimentech bylo tudíž pracováno pouze s ABS normalizací signálu.

**Tabulka 4.5:** Porovnání Acc pro různé normalizace signálů

Spektrum	Acc [%]
/10	79.20
/5	77.78
Výkonové	77.78



**Tabulka 4.6:** Porovnání WADA SNR out pro různé normalizace spekter

Spektrum	SNR in [dB]	-5	0	5	10	15	20
		WADA SNR in[dB]	-0.92	0.59	4.28	8.53	12.92
/5	WADA SNR out[dB]	20.59	28.39	33.42	35.39	36.10	35.71
/10	WADA SNR out[dB]	17.39	24.46	31.08	34.65	35.83	35.95
Výkonové	WADA SNR out[dB]	8.88	15.12	21.58	26.71	30.93	33.86

**Tabulka 4.7:** Porovnání PEQS out pro různé normalizace spekter

Spektrum	SNR in [dB]	-5	0	5	10	15	20
		PESQ in	1.76	2.05	2.42	2.87	3.32
/5	PESQ out	1.90	2.33	2.82	3.30	3.69	3.96
/10	PESQ out	1.85	2.24	2.72	3.20	3.61	3.90
Výkonové	PESQ out	1.88	2.27	2.75	3.23	3.62	3.87

Z tabulek vyplývá, že nejlepší je využití amplitudových spekter, která jsou přeškálována vydělením pěti. Přeškálování vydělením deseti přináší také velmi dobré výsledky. Při subjektivním porovnání poslechem výstupů sítě, není mezi těmito přístupy slyšitelný téměř žádný rozdíl. Výkonové spektrum je z hlediska hodnot PESQ pro nižší SNR srovnatelné s oběma předchozími přístupy. Z hlediska potlačení aditivních šumů (WADA SNR) nedosahuje tak dobrých výsledků.

### 4.3.3 Velikost dávky

Další byla testována velikost dávky při učení neuronové sítě. Byly otestovány standardní velikosti dávky a to: 32, 64, 128 a 256. Testovací dataset byl před učením DNN promíchán, pro lepší generalizaci. Učení probíhalo po dobu 30 epoch. Po této době již síť nevykazovala zlepšení v učení. Tabulka 4.8 zobrazuje porovnání Acc pro dané velikosti dávky. Tabulka 4.9 a 4.10 pak ukazují výsledky naučené DNN na testovacím datasetu.

**Tabulka 4.8:** Porovnání Acc pro různé velikosti dávky

Velikost dávky	Acc [%]
32	60.50
64	79.19
128	79.20
256	78.21

**Tabulka 4.9:** Porovnání WADA SNR out pro jednotlivé velikosti dávky

Velikost dávky	SNR in [dB]	-5	0	5	10	15	20
		WADA SNR in[dB]	-0.92	0.59	4.28	8.53	12.92
32	WADA SNR out[dB]	8.94	12.21	14.80	16.42	17.13	17.17
64	WADA SNR out[dB]	19.54	26.94	31.84	34.32	35.66	36.00
128	WADA SNR out[dB]	17.39	24.46	31.08	34.65	35.83	35.95
256	WADA SNR out[dB]	13.50	20.41	28.03	35.38	38.89	38.66

**Tabulka 4.10:** Porovnání PEQS out pro jednotlivé velikosti dávky

Velikost dávky	SNR in [dB]	-5	0	5	10	15	20
		PESQ in	1.76	2.05	2.42	2.87	3.32
32	PESQ out	1.64	1.89	2.20	2.53	2.86	3.12
64	PESQ out	1.85	2.27	2.75	3.23	3.63	3.91
128	PESQ out	1.85	2.24	2.72	3.20	3.61	3.90
256	PESQ out	1.80	2.19	2.66	3.15	3.57	3.87

Z tabulek vyplývá, že modely s velikostí dávky 64, 128 a 256 jsou srovnatelné, kdežto model s velikostí dávky 32 je o mnoho horší.

#### 4.3.4 Velikost okolí

Standardní praxí při metodách zvýrazňování řeči je využití centrálního spektra +  $n$  okolních spekter na každou stranu na vstupu DNN. Největší vliv toto má při odstraňování konvolučních šumů, jelikož dozvuky mají obvykle trvání delší, než pojme jeden segment. U odstraňování aditivních šumů, byl předpokládáný přínos okolí k zachování harmonické struktury řeči na výstupu. Byly otestovány 3 velikosti okolí: 0, 5 a 10 tj. 1 segment, 11 segmentů a 21 segmentů. Tabulky 4.11, 4.12 a 4.13 zobrazují porovnání výsledků DNN pro jednotlivé velikosti okolí.

**Tabulka 4.11:** Porovnání Acc pro různé velikosti okolí

Velikost okolí	Acc [%]
0	75.97
5	79.20
10	32.46

Jak je z tabulky patrné, nejlepšího výsledku dosáhla síť s okolím 5. Okolí 10, tj. 21 segmentů, tj. 2709 vzorků bylo pro tuto DNN již příliš velké a síť se nepodařilo namapovat vstupy na výstupy.

**Tabulka 4.12:** Porovnání WADA SNR out pro jednotlivé velikosti okolí

Velikost okolí	SNR in [dB]	-5	0	5	10	15	20
		WADA SNR in[dB]	-0.92	0.59	4.28	8.53	12.92
0	WADA SNR out[dB]	10.8	16.33	23.97	31.09	36.46	38.32
5	WADA SNR out[dB]	17.39	24.46	31.08	34.65	35.83	35.95

**Tabulka 4.13:** Porovnání PEQS out pro jednotlivé velikosti okolí

Velikost okolí	SNR in [dB]	-5	0	5	10	15	20
		PESQ in	1.76	2.05	2.42	2.87	3.32
0	PESQ out	1.72	2.06	2.51	3.00	3.45	3.79
5	PESQ out	1.85	2.24	2.72	3.20	3.61	3.90

Z tabulek vyplývá, že přidružení přiměřeně velkého okolí má pozitivní vliv na výkon sítě. U sítě s okolím 5 jsou vyšší hodnoty PESQ pro všechna SNR in. WADA SNR vykazuje obecně také lepší hodnoty.

### 4.3.5 Promíchání dat

Při vytváření trénovací databáze jsou spektra resp. spektra s okolím do ní ukládána tak jak jsou načítány soubory se spektry jednotlivých promluv, tj. spektra z jedné promluvy jsou v databázi u sebe. Cílem této části experimentu bylo zjistit, zda promíchání datasetu přinese zlepšení. Teoretickým předpokladem bylo, že promíchání by mohlo pomoci s generalizací. Pro otestování byl zvolen model s velikostí dávky 256.

**Tabulka 4.14:** Porovnání Acc pro nepromíchaný a promíchaný dataset

Promícháno	Acc [%]
Ne	76.48
Ano	78.65

**Tabulka 4.15:** Porovnání WADA SNR out pro nepromíchaný a promíchaný dataset

Promícháno	SNR in [dB]	-5	0	5	10	15	20
		WADA SNR in[dB]	-0.92	0.59	4.28	8.53	12.92
Ne	WADA SNR out[dB]	13.91	19.92	26.17	31.55	34.74	34.74
Ano	WADA SNR out[dB]	13.50	20.41	28.03	35.38	38.89	38.66

**Tabulka 4.16:** Porovnání PESQ out pro nepromíchaný a promíchaný dataset

Promícháno	SNR in [dB]	-5	0	5	10	15	20
		PESQ in	1.76	2.05	2.42	2.87	3.32
Ne	PESQ out	1.82	2.20	2.67	3.15	3.58	3.88
Ano	PESQ out	1.80	2.19	2.66	3.15	3.57	3.87

Z tabulek je patrné, že promíchaná data vykazují velmi podobné výsledky jako nepromíchaná. Jediným rozdílem je v průměru o několik decibelů větší WADA SNR u promíchaných. Data není nutné při procesu učení DNN promíchat, což může u větších datasetů prodloužit výpočetní čas.

### 4.3.6 Zhodnocení experimentu

Jak je patrné z tabulek u jednotlivých experimentů, DNN, která byla trénována na smíšených signálech v rozmezí 8 dB až 20 dB, je schopna velmi dobře ve všech konfiguracích potlačovat aditivní šумы v řečových signálech, i pro SNR na které nebyla trénována. Srozumitelnost zvýrazněných signálů je taktéž dobrá. U signálů smíchaných při -5 dB a 0 dB se příležitostně v malé míře vyskytují reziduální zbytky šumu a signál je zkreslený, čemuž odpovídají i hodnoty PESQ, ale i tak si síť vede nad očekávání dobře i ve srovnání s jinými publikovanými modely. DNN velmi dobře zpracovala i mužské i ženské hlasy.

Nejlepší nastavení pro DNN byla velikost dávky 64, normování signálu abs škálováním a normování spekter škálováním /5. Dále v textu bude nazýváno Model64. Téměř schodných výsledků dosahoval model s velikostí dávky 128.

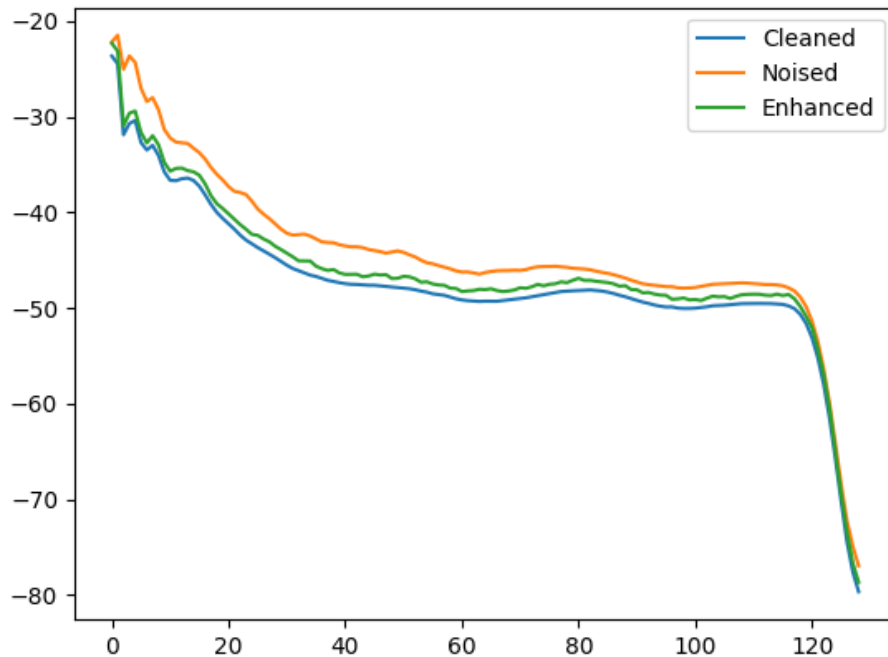
Tabulka 4.17 ukazuje srovnání mezi výsledky modelu Model64 a mezi výsledky prezentovanými v práci [3] a [7].

**Tabulka 4.17:** Porovnání PESQ out pro model Model64 a modely prezentovanými v [7] a [3]

Model	SNR in [dB]	-5	0	5	10	15	20
Model publikovaný v [7]	PESQ out	1.59	1.92	2.27	2.53	2.93	3.23
Model publikovaný v [3]	PESQ out	1.74	2.10	2.46	2.80	3.10	3.39
Model 64	PESQ out	1.85	2.27	2.75	3.23	3.63	3.91

Zajímavým zjištěním plynoucím z tabulky je, že model, který byl učen na smíchaných signálech v rozmezí 8 dB - 20 dB vykazuje podobné výsledky při SNR in  $-5 \div 5$  jako modely, které byly i při těchto SNR učeny.

Obrázek 4.2 zobrazuje průměrné výkonové spektrum napočítané přes celé databáze (zašuměné, referenční čisté, upravené). Z obrázku je patrné, že čisté a upravené signály mají průměrné spektrum velmi podobné, jejich tvar je téměř totožný, a jejich průměrný rozdíl je 1dB.

**Obrázek 4.2:** Průměrné spektrum zašuměných, referenčních a zvýrazněných signálů. Pro lepší srozumitelnost převedeno na spektra výkonová.

Výsledky PESQ a WADA SNR pro jednotlivé soubory části testovacího datasetu je možné nalézt v Příloze A. Testovací datasety, použité kódy i jednotlivé modely jsou k dispozici na [https://noel.fel.cvut.cz/BP\\_Jiri\\_Smid/](https://noel.fel.cvut.cz/BP_Jiri_Smid/)

## 4.4 Experiment 2: 3 skryté vrstvy po 1500 neuronch a T1500+GEN\_NOISES

Na základě poznatků z předchozího experimentu byl připraven další experiment s komplexnějším datasetem T1500+COMPLEX. V tomto případě, kdy bylo do datasetu přidáno přes sto nových šumů se ukázalo, že přístup ke každému datasetu musí být individuální. Stejný postup míchání signálu se šumy, jako v případě experimentu 1 se ukázal jako nedostačující. Data nebyla dostatečně komplexní a neuronová síť nebyla schopna se správně naučit. Maximální Acc byla 37.37%.

Dalším krokem tedy bylo zajištění rozmanitějšího datasetu. Z tohoto důvodu bylo přistoupeno k míšení signálů se šumy při více SNR, tj. Signál se šumem byl smíchán a uložen pro -5, 0, 5, 10, 15, 20 dB. Finální dataset tedy obsahoval 6 x 1551 promluv. Toto zvětšení datasetu vedlo k pokroku v učení neuronové sítě a po sérii experimentů s optimalizací rychlosti učení bylo dosaženo Acc 58.88 %. Tabulky 4.18 a 4.19 ukazují výsledky modelu na testovacím datasetu.

**Tabulka 4.18:** WADA SNR modelu s mícháním při více SNR

SNR in [dB]	-5	0	5	10	15	20
WADA SNR in [dB]	2.78	2.95	9.31	12.96	16.65	20.12
WADA SNR out [dB]	17.35	20.19	24.00	28.10	29.95	30.75

**Tabulka 4.19:** PESQ modelu s mícháním při více SNR

SNR in [dB]	-5	0	5	10	15	20
PESQ in	1.56	1.80	2.09	2.44	2.84	3.29
PESQ out	1.56	1.74	1.94	2.12	2.28	2.42

Z tabulek vyplývá, že k potlačení šumu dochází, ale nikoliv optimálně, PESQ je velmi nízký. Poslech výstupů odhalil, že model potlačuje šumy v pauzách mezi řečí, ale v řečových úsecích zůstávají lehce potlačené reziduální ruchy. Signál je navíc lehce zkreslen, což se projevuje v hodnotách PESQ.

Na základě dvou předchozích pokusů bylo přistoupeno k dalšímu experimentu s míšením signálů a šumů. V tomto případě dochází k míšení jedné promluvy s 20 šumy při 6 hodnotách SNR. Toto by mělo zvětšit variabilitu a zastoupení šumů v datasetu, a tím zlepšit výsledky sítě.

Výsledky tohoto experimentu jsou v současné době ve fázi zpracování.

## 5 Závěr

Tato práce se zabývala možností využití plně propojených dopředných sítí za účelem zvýrazňování řečových signálů tj. potlačování aditivních šumů v řečových signálech. V rámci experimentální části byly vyzkoušeny různé velikosti a konfigurace dopředných neuronových sítí, a různé formáty vstupních dat. Z výsledků práce je patrné, že tyto DNN dokáží velmi dobře potlačit i nestacionární šумы a stále zachovat harmonickou složku řeči. Toto je možné vidět z hodnot PESQ. Výsledky experimentů nadále ukazují, že není nutné trénovat DNN pro všechna SNR, která obsahuje evaluační dataset. DNN natrénovaná na datech, která vnikla smícháním signálu se šумы v rozmezí  $8 \div 20$  dB, podávala srovnatelné výsledky i pro nižší SNR jako sítě natrénované i pro tato nižší SNR. Práce dále ukázala, že není nutné spektra normalizovat za pomoci MVN, jak je uváděno v mnoha pracech, postačí vhodné přeškálování, tak, aby data byla v rozsahu strmé části aktivační funkce.

Výsledná neuronová síť může najít uplatnění jako preprocessingová část nástrojů pro rozpoznání řeči, či řečníka, ale kvalita výstupu je natolik dobrá, že může posloužit i k poslechu.

Výsledky z této práce bych rád využil pro další výzkum v oblasti vylepšení řečových signálů, nejen za pomoci plně propojených dopředných sítí, ale i s pomocí rekurentních a konvolučních neuronových sítí. Využití konvolučních vrstev by mohlo být výhodné pro zachování harmonické struktury řeči i při podmínkách, které by byly pro dopřednou síť velmi obtížné. Rekurentní neuronové sítě by díky paměti mohly pomoci s potlačováním konvolučních šumů, tj. ech a dozvuků. Dále by tyto struktury mohly pomoci při budování reálného nástroje pro veřejné používání, jelikož datasety pro učení DNN budou muset být mnohem více heterogenní tj. mít větší zastoupení řečníků i šumů, což by pro dopřednou síť mohlo být problematické i se zvýšením počtu neuronů.

V další práci bych se tedy rád zaměřil na větší realističnost vstupních dat a do trénovacích datasetů přidal více druhů nestacionárních šumů a především zahrnul více řečníků mluvících více jazyky. Další možnost rozšíření vidím v přidání detektoru řečové aktivity. Další možností také je využití fáze resp. komplexní reprezentace spektra.





# Seznam obrázků

2.1	Ilustrace VAD na spektrogramu. Šrafovaně jsou vyznačeny segmenty obsahující pouze šum . . . . .	6
2.2	Schéma metody spektrálního odečítání . . . . .	7
2.3	Čtyři základní aktivační funkce . . . . .	10
2.4	Schéma metody využité v práci . . . . .	13
2.5	Ilustrace zvýraznění řeči pomocí DNN . . . . .	14
3.1	Schéma metody využité v práci . . . . .	17
4.1	Ilustrace mapování zašuměných spekter na čistá. Parametr area odpovídá počtu spekter (centrální + okolí), která vstupují do sítě . . . . .	26
4.2	Průměrné spektrum zašuměných, referenčních a zvýrazněných signálů. Pro lepší srozumitelnost převedeno na spektra výkonová. . . . .	33

\*

# Seznam tabulek

4.1	Porovnání SNR a WADA SNR . . . . .	27
4.2	Porovnání Acc pro různé normalizace souborů . . . . .	28
4.3	Porovnání WADA SNR out pro různé normalizace signálů . . . . .	28
4.4	Porovnání PEQS out pro různé normalizace signálů . . . . .	28
4.5	Porovnání Acc pro různé normalizace signálů . . . . .	28
4.6	Porovnání WADA SNR out pro různé normalizace spekter . . . . .	29
4.7	Porovnání PEQS out pro různé normalizace spekter . . . . .	29
4.8	Porovnání Acc pro různé velikosti dávky . . . . .	29
4.9	Porovnání WADA SNR out pro jednotlivé velikosti dávky . . . . .	30
4.10	Porovnání PEQS out pro jednotlivé velikosti dávky . . . . .	30
4.11	Porovnání Acc pro různé velikosti okolí . . . . .	30
4.12	Porovnání WADA SNR out pro jednotlivé velikosti okolí . . . . .	31
4.13	Porovnání PEQS out pro jednotlivé velikosti okolí . . . . .	31
4.14	Porovnání Acc pro nepromíchaný a promíchaný dataset . . . . .	31
4.15	Porovnání WADA SNR out pro nepromíchaný a promíchaný dataset . . . . .	32
4.16	Porovnání PESQ out pro nepromíchaný a promíchaný dataset . . . . .	32
4.17	Porovnání PESQ out pro model Model64 a modely prezentovanými v [7] a [3] . . . . .	33
4.18	WADA SNR modelu s mícháním při více SNR . . . . .	34
4.19	PESQ modelu s mícháním při více SNR . . . . .	34

\*

# Bibliografie

- [1] „Robust Signal-to-Noise Ratio Estimation Based on Waveform Amplitude Distribution Analysis“. (), URL: <http://www.cs.cmu.edu/~robust/Papers/KimSternIS08.pdf> (cit. 26.02.2023).
- [2] Z. Bai a X.-L. Zhang, „Speaker recognition based on deep learning: An overview“, *Neural Networks*, roč. 140, s. 65–99, 2021, ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2021.03.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0893608021000848>.
- [3] L.-R. D. Y. Xu J. Du a C.-H. Lee, „A Regression Approach to Speech Enhancement Based on Deep Neural Networks“, 1, sv. 23, led. 2015, s. 7–19. DOI: 10.1109/TASLP.2014.2364452.
- [4] J. Tchorz a B. Kollmeier, „SNR estimation based on amplitude modulation analysis with applications to noise suppression“, *IEEE Transactions on Speech and Audio Processing*, roč. 11, č. 3, s. 184–192, 2003. DOI: 10.1109/TSA.2003.811542.
- [5] S. Tamura, „An analysis of a noise reduction neural network“, *International Conference on Acoustics, Speech, and Signal Processing*,, 2001–2004 vol.3, 1989.
- [6] G. E. Hinton, S. Osindero a Y.-W. Teh, „A Fast Learning Algorithm for Deep Belief Nets“, *Neural Computation*, roč. 18, č. 7, s. 1527–1554, čvc. 2006, ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527.
- [7] L.-R. D. Y. Xu J. Du a C.-H. Lee, „An experimental study on speech enhancement based on deep neural networks“, 1, sv. 21, led. 2014, s. 65–68. DOI: 10.1109/LSP.2013.2291240.
- [8] M. Mimura, S. Sakai a T. Kawahara, „Reverberant speech recognition combining deep neural networks and deep autoencoders augmented with a phone-class feature“, *EURASIP Journal on Applied Signal Processing*, roč. 2015, 62, s. 62, pros. 2015. DOI: 10.1186/s13634-015-0246-6.
- [9] „A Gentle Introduction to the Rectified Linear Unit (ReLU)“. (), URL: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> (cit. 23.03.2023).
- [10] „Loss Functions and Their Use in Neural Networks“. (), URL: <https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9> (cit. 23.03.2023).
- [11] „Applied Deep Learning - Part 3: Autoencoders“. (), URL: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798> (cit. 16.04.2023).
- [12] „Understand the Dynamics of Learning Rate on Deep Learning Neural Networks“. (), URL: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/> (cit. 23.03.2023).
- [13] „How to Control Stability of Training Neural Networks with Gradient Descent Batch Size“. (), URL: <https://machinelearningmastery.com/how-to-control-the-speed-and-stability-of-training-neural-networks-with-gradient-descent-batch-size/> (cit. 23.03.2023).

- 
- [14] „Various Optimization Algorithms For Training Neural Network“. (), URL: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6> (cit. 16.04.2023).
- [15] „NumPy“. (), URL: <https://numpy.org/> (cit. 26.02.2023).
- [16] „SciPy“. (), URL: <https://scipy.org/> (cit. 26.02.2023).
- [17] „Librosa“. (), URL: <https://librosa.org/> (cit. 26.02.2023).
- [18] „Soundfile“. (), URL: <https://github.com/bastibe/python-soundfile> (cit. 26.02.2023).
- [19] „Cmath“. (), URL: <https://docs.python.org/3/library/cmath.html> (cit. 26.02.2023).
- [20] „TensorFlow“. (), URL: <https://www.tensorflow.org/> (cit. 26.02.2023).
- [21] „Keras“. (), URL: <https://keras.io/> (cit. 26.02.2023).
- [22] „Metacentrum“. (), URL: <https://www.metacentrum.cz/cs/> (cit. 26.02.2023).
- [23] „Metacentrum“. (), URL: <https://metavo.metacentrum.cz/cs/about/index.html> (cit. 26.02.2023).
- [24] „Mathematics and Signal Processing for Biomechanics - Windowing“. (), URL: <https://www1.udel.edu/biology/rosewc/kaap686/notes/windowing.html> (cit. 26.02.2023).
- [25] „Which is faster to store data using Python, CSV or text?“ (), URL: <https://dev.to/marianvelani/who-is-faster-to-store-data-using-python-csv-or-text-25ea> (cit. 26.02.2023).
- [26] „Python CSV library“. (), URL: <https://docs.python.org/3/library/csv.html> (cit. 26.02.2023).
- [27] C. K. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan a J. Gehrke, „A Scalable Noisy Speech Dataset and Online Subjective Test Framework“, *Proc. Interspeech 2019*, s. 1816–1820, 2019.

# Přílohy

## A Ukázka WADA SNR a PESQ pro část testovacího datasetu k T1500+CAR

SNR in PESQ	Noised						Enhanced					
	-5,00	0,00	5,00	10,00	15,00	20,00	-5,00	0,00	5,00	10,00	15,00	20,00
T16102Z2.wav	1,61	1,87	2,28	2,78	3,32	3,85	1,47	1,82	2,28	2,78	3,25	3,57
T16112Z2.wav	1,96	2,34	2,76	3,16	3,52	3,90	1,89	2,35	2,88	3,41	3,72	3,96
T17202Z1.wav	1,40	1,64	2,00	2,48	3,09	3,72	1,77	2,28	2,75	3,26	3,66	3,96
T16112S9.wav	1,75	2,03	2,40	2,83	3,26	3,72	1,45	1,81	2,31	2,98	3,61	3,97
T10100Z6.wav	1,77	2,09	2,47	2,88	3,40	3,86	2,37	2,94	3,44	3,81	4,09	4,28
T16112S1.wav	1,88	2,02	2,25	2,65	3,19	3,75	1,49	1,83	2,31	2,95	3,56	3,91
T16106Z5.wav	1,25	1,42	1,56	1,81	2,23	2,77	1,54	1,80	2,19	2,65	3,13	3,55
T15111S9.wav	1,90	2,23	2,59	3,04	3,46	3,85	1,91	2,19	2,55	2,76	3,56	3,96
T14208S4.wav	1,86	2,29	2,75	3,27	3,71	3,99	1,83	2,16	2,63	3,11	3,56	3,89
T17101S0.wav	1,79	2,11	2,59	3,10	3,67	4,06	1,69	2,13	2,62	3,19	3,76	4,09
T16114Z2.wav	1,38	1,60	1,94	2,39	2,94	3,51	1,55	1,93	2,41	2,92	3,39	3,71
T16207Z3.wav	1,57	1,87	2,28	2,80	3,31	3,78	2,37	2,85	3,41	3,94	4,20	4,35
T10100Z1.wav	1,67	1,97	2,34	2,77	3,15	3,57	2,26	2,75	3,30	3,63	3,99	4,19
T16208S3.wav	2,01	2,37	2,72	3,15	3,61	3,99	2,12	2,59	3,06	3,57	3,95	4,19
T15101S3.wav	1,84	2,20	2,66	3,12	3,55	3,95	1,53	1,91	2,43	3,14	3,78	4,06
T16115S7.wav	1,73	2,03	2,40	2,89	3,26	3,67	1,89	2,35	2,87	3,32	3,68	3,95
T15101S8.wav	2,04	2,42	2,88	3,36	3,85	4,19	1,64	2,01	2,52	3,13	3,68	4,04
T16210Z1.wav	1,85	2,04	2,34	2,75	3,21	3,68	1,85	2,20	2,67	3,20	3,66	4,00
T14112S7.wav	1,55	1,87	2,33	2,86	3,45	3,95	2,08	2,52	3,11	3,51	3,79	3,98
T10100S0.wav	1,75	2,12	2,54	2,96	3,47	3,99	1,85	2,21	2,69	3,12	3,45	3,76
T17202S1.wav	2,01	2,36	2,78	3,29	3,74	4,09	1,40	1,71	2,15	2,68	3,22	3,68
T15111S2.wav	1,91	2,26	2,66	3,12	3,62	4,03	1,87	2,17	2,61	3,20	3,62	3,94
T15217S3.wav	1,96	2,26	2,64	3,10	3,60	3,92	1,39	1,59	1,82	2,16	2,45	2,77
T10100S4.wav	1,85	2,15	2,53	3,03	3,63	4,04	1,98	2,31	2,64	2,97	3,30	3,54
T17101S6.wav	1,65	2,08	2,59	3,10	3,71	4,14	1,31	1,67	2,20	2,68	3,16	3,55
T10100S2.wav	1,76	2,05	2,43	2,80	3,16	3,47	1,95	2,36	2,79	3,19	3,55	3,83
T16117Z7.wav	1,69	1,98	2,32	2,69	3,14	3,63	1,48	1,84	2,31	2,80	3,23	3,60
T15114S0.wav	2,01	2,42	2,91	3,35	3,75	4,07	2,13	2,58	3,01	3,50	3,80	4,03
T16204Z2.wav	1,61	1,97	2,42	2,95	3,36	3,79	2,12	2,71	3,23	3,68	3,98	4,18
T15203S2.wav	1,70	2,06	2,48	3,00	3,58	3,98	1,99	2,34	3,14	3,69	4,05	4,09
T15101S7.wav	1,69	1,98	2,37	2,79	3,18	3,63	1,59	2,00	2,62	3,30	3,73	3,99
T17201S9.wav	1,50	1,81	2,22	2,72	3,35	3,87	2,15	2,64	3,07	3,38	3,64	3,96
T16112Z0.wav	1,88	2,18	2,54	2,92	3,50	3,97	1,69	2,08	2,55	2,97	3,25	3,54
T15212S6.wav	1,29	1,51	1,82	2,25	2,73	3,21	2,14	2,82	3,46	3,89	4,11	4,25
T16110Z4.wav	2,02	2,32	2,70	3,20	3,73	4,15	1,90	2,34	2,87	3,38	3,85	4,08
T15211S7.wav	1,43	1,68	2,06	2,61	3,32	3,89	1,86	2,35	2,90	3,41	3,86	4,12
T14301S7.wav	1,72	2,07	2,52	3,07	3,63	4,08	1,89	2,26	2,75	3,28	3,74	3,98
T16203Z9.wav	1,65	1,85	2,16	2,57	3,02	3,16	2,02	2,48	2,92	3,35	3,81	4,14
T16205S2.wav	1,68	1,99	2,36	2,84	3,39	3,87	2,31	2,78	3,29	3,70	4,06	4,28
T15223S2.wav	1,96	2,36	2,85	3,38	3,87	4,20	1,90	2,33	2,85	3,28	3,60	3,94
T10100S9.wav	1,42	1,62	1,95	2,41	2,94	3,47	1,92	2,38	2,81	3,23	3,58	3,81
T14117S8.wav	1,59	1,88	2,29	2,74	3,27	3,75	1,89	2,41	2,91	3,34	3,67	3,90
T14107S2.wav	1,42	1,65	2,01	2,46	2,95	3,58	1,29	1,38	1,63	2,03	2,44	2,89
T16118S8.wav	1,40	1,67	2,03	2,47	2,95	3,44	1,63	1,98	2,42	2,90	3,33	3,69
T15210S7.wav	1,52	1,69	1,93	2,30	2,75	3,27	2,36	2,94	3,36	3,77	4,03	4,20
T14108S3.wav	1,71	2,02	2,40	2,82	3,31	3,78	2,28	2,76	3,34	3,77	4,02	4,21
T10100Z3.wav	1,95	2,23	2,60	3,01	3,39	3,72	1,89	2,40	2,92	3,47	3,90	4,15
T16207S2.wav	1,37	1,55	1,82	2,21	2,67	3,12	2,18	2,70	3,17	3,58	3,91	4,17
T16118Z8.wav	1,72	2,06	2,48	3,00	3,59	3,99	1,89	2,32	2,87	3,30	3,62	3,95
T14119S7.wav	1,47	1,74	2,13	2,68	3,29	3,84	2,08	2,49	2,97	3,41	3,69	3,90
T14101S6.wav	1,75	1,97	2,29	2,75	3,24	3,73	1,54	1,97	2,41	2,91	3,44	3,60
Prumerne PESQ	1,70	2,00	2,38	2,84	3,33	3,78	1,85	2,27	2,75	3,23	3,63	3,91

SNR in	Noised						Enhanced					
	-5,00	0,00	5,00	10,00	15,00	20,00	-5,00	0,00	5,00	10,00	15,00	20,00
WADA SNR												
T16102Z2.wav	-4,45	0,26	4,58	9,27	14,17	18,37	19,23	28,80	39,47	54,60	87,70	88,76
T16112Z2.wav	4,89	5,87	8,73	13,31	19,17	26,55	30,80	48,19	44,88	36,75	32,44	30,21
T17202Z1.wav	-7,49	-0,72	4,41	9,51	14,05	18,19	21,09	25,87	32,10	36,67	38,10	37,42
T16112S9.wav	3,27	4,90	8,08	12,36	17,37	22,75	8,25	15,58	23,59	27,07	27,01	26,20
T10100Z6.wav	-2,40	-0,47	3,77	7,78	11,41	14,37	21,57	24,17	24,38	23,84	23,02	22,51
T16112S1.wav	0,64	3,19	6,93	10,88	14,60	17,06	15,29	20,86	22,06	22,27	21,93	21,10
T16106Z5.wav	-0,29	1,89	4,88	8,39	11,73	14,37	12,65	29,92	60,96	59,76	50,40	44,90
T15111S9.wav	-6,00	-0,44	4,58	9,42	14,42	19,38	17,15	20,05	21,73	21,72	20,61	19,70
T14208S4.wav	5,39	6,13	8,84	12,56	16,29	20,54	26,20	33,37	41,97	49,01	50,76	47,80
T17101S0.wav	2,18	3,53	5,97	8,80	11,79	13,74	12,83	13,77	13,23	12,62	11,74	11,59
T16114Z2.wav	-0,11	2,61	6,41	10,68	14,95	18,65	16,62	24,10	27,30	31,13	38,87	50,50
T16207Z3.wav	-20,00	-2,22	3,59	8,90	12,66	15,76	17,16	20,10	22,07	21,72	20,79	19,86
T10100Z1.wav	4,26	6,34	10,31	15,29	21,94	29,53	15,53	22,27	26,25	26,45	24,69	23,80
T16208S3.wav	2,34	4,27	8,07	12,63	17,76	23,14	20,32	20,24	19,26	17,27	16,57	16,00
T15101S3.wav	-0,49	1,21	5,08	9,71	13,75	17,66	6,45	12,34	16,71	18,57	18,40	18,07
T16115S7.wav	-10,04	-0,91	4,66	10,28	15,95	22,45	16,18	22,01	26,64	31,95	37,49	42,35
T15101S8.wav	5,02	6,00	8,42	11,71	14,70	16,68	15,31	24,20	26,33	25,46	24,82	23,58
T16210Z1.wav	1,46	3,95	7,80	12,71	18,28	24,34	15,86	20,15	21,72	22,36	22,43	21,54
T14112S7.wav	2,22	3,80	6,59	10,12	14,11	17,82	24,25	32,96	34,66	34,88	33,96	33,33
T10100S0.wav	-2,94	0,94	4,95	9,14	12,45	15,12	15,54	27,06	34,64	38,53	42,48	48,30
T17202S1.wav	6,02	7,20	9,67	13,32	17,37	20,96	22,33	27,71	30,60	32,43	31,45	28,08
T15111S2.wav	2,59	4,61	7,56	11,21	14,63	17,16	20,95	25,04	23,22	20,82	19,16	18,35
T15217S3.wav	0,37	2,88	6,70	11,03	15,08	18,35	21,03	25,96	29,28	30,49	31,97	31,22
T10100S4.wav	0,63	2,12	4,88	8,19	11,19	13,01	18,12	40,12	56,99	73,28	100,00	100,00
T17101S6.wav	3,27	5,52	8,31	12,21	16,67	21,32	14,78	12,43	12,77	11,72	10,55	9,53
T10100S2.wav	-20,00	-1,18	4,63	10,25	15,69	21,86	17,83	35,12	49,88	60,04	73,21	94,70
T16117Z7.wav	-0,11	3,32	7,58	12,72	18,33	24,63	12,53	21,46	29,92	34,88	36,59	37,81
T15114S0.wav	-9,41	-0,55	4,96	10,25	14,95	18,68	30,80	35,66	40,76	41,54	39,07	35,49
T16204Z2.wav	0,39	3,73	7,98	12,04	15,81	18,50	22,88	23,40	25,28	27,29	27,73	27,49
T15203S2.wav	-0,53	2,49	6,53	11,11	15,42	19,65	15,35	25,07	27,73	27,30	26,23	24,96
T15101S7.wav	-20,00	-2,34	3,77	9,48	14,81	20,11	7,45	13,32	19,11	19,71	19,42	19,21
T17201S9.wav	-1,63	0,98	3,36	6,36	8,59	9,75	43,02	53,90	70,57	100,00	100,00	100,00
T16112Z0.wav	-4,82	-0,28	5,01	10,26	15,04	18,85	15,98	29,76	40,33	43,49	43,61	46,04
T15212S6.wav	-0,07	0,31	2,86	5,43	7,00	7,78	14,37	16,95	17,52	17,49	16,51	15,87
T16110Z4.wav	1,84	4,01	8,06	12,69	17,35	21,99	15,65	23,35	30,38	33,06	33,21	32,30
T15211S7.wav	-5,16	-0,92	4,06	8,49	12,51	15,45	15,51	18,30	21,59	21,17	20,33	19,89
T14301S7.wav	-1,37	2,13	6,28	10,61	15,27	19,23	12,17	21,60	31,69	36,96	38,89	37,97
T16203Z9.wav	-1,21	2,26	5,95	10,54	14,57	18,22	16,36	21,17	20,45	19,85	19,03	18,69
T16205S2.wav	-0,04	3,07	7,32	12,16	17,28	22,42	19,74	26,20	31,03	34,48	34,82	33,60
T15223S2.wav	6,35	7,35	10,29	13,84	17,07	19,02	24,09	30,76	36,05	37,63	38,21	36,29
T10100S9.wav	0,61	3,28	7,12	11,88	17,13	23,15	43,99	65,57	64,94	100,00	100,00	100,00
T14117S8.wav	-3,51	0,51	5,66	10,39	15,83	21,25	21,88	28,72	30,90	28,77	27,69	25,59
T14107S2.wav	-2,83	0,81	5,29	9,91	14,40	18,90	25,28	31,67	28,42	23,94	22,40	21,11
T16118S8.wav	-3,94	0,16	4,83	9,70	14,19	18,64	21,03	26,32	25,54	26,92	26,35	26,28
T15210S7.wav	-2,50	0,66	5,61	10,25	15,27	20,64	27,25	30,43	28,99	28,71	27,96	26,61
T14108S3.wav	-2,86	0,74	5,53	9,63	13,73	17,08	16,89	19,66	21,42	21,65	21,60	20,49
T10100Z3.wav	-6,52	-0,42	4,95	10,31	16,05	21,88	26,29	29,38	27,98	24,88	23,87	22,79
T16207S2.wav	5,18	6,71	9,19	12,02	15,31	18,77	24,56	28,90	28,15	26,20	24,73	23,28
T16118Z8.wav	1,00	3,39	7,19	11,65	15,75	18,73	21,95	26,61	30,89	33,08	32,91	31,97
T14119S7.wav	-7,03	-0,64	4,13	8,21	11,76	14,50	22,02	50,49	85,74	76,22	83,79	100,00
T14101S6.wav	-3,65	0,04	4,76	8,83	12,56	15,27	15,97	23,09	25,75	23,85	23,41	23,00
Prumerne WADA SNR	-1,79	2,20	6,21	10,56	14,87	18,87	19,54	26,94	31,84	34,32	35,66	36,00