Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

Diplomová práce

Detekce hlasivkových pulsů v řečovém signálu pomocí strojového učení

Plzeň 2020 Bc. Michal Vraštil

Místo této strany bude zadání práce.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 30. června 2020

Bc. Michal Vraštil

Poděkování

Chtěl bych poděkovat svému vedoucímu diplomové práce panu Doc. Ing. Jindřichu Matouškovi, Ph.D. za jeho pomoc, odborné rady a připomínky při zpracování této práce. Dále bych chtěl poděkovat svým rodičům a přítelkyni za obětavou pomoc a podporu, kterou mi při psaní této práce poskytli.

Abstract

The topic of this diploma thesis is the detection of glottal closure instants from the speech signal using machine learning methods. It aims to improve the success of the classification of the initial algorithm, especially by adding new features and finding other suitable methods of speech signal preprocessing. The introduction of this work briefly describes the physiological production of speech and glottal closure instants, their computer processing and the importance and benefits of their accurate detection. Subsequently, the initial algorithm is introduced and the reader is introduced to the process of finding new features and new methods of preprocessing. The main benefit for the success of the classification is achieved by the implementation of the Context aware classifier, which is then used for the rest of the work. Finally, the improved algorithm is compared with the initial algorithm. Furthermore, its success is verified on data that have passed through the simulated telephone channel and on data that have been modified in the same way with added white noise.

Keywords

glottal closure instant, pitch mark, detection, classification, extreme gradient boosting, convolutional neural net, python, context aware classifier

Abstrakt

Tématem této diplomové práce je detekce hlasivkových pulsů v řečovém signálu pomocí metod strojového učení. Klade si za cíl vylepšit úspěšnost klasifikace výchozího algoritmu, a to především přidáním nových příznaků, nalezením dalších vhodných metod předzpracování řečového signálu a implementací kontextového klasifikátoru. V úvodu této práce je stručně popsána fyziologická tvorba řečového signálu a hlasivkových pulsů, jejich zpracování počítačem a význam a přínos jejich přesné detekce. Následně je představen výchozí algoritmus a čtenář je seznámen s postupem nalezení nových příznaků a nových metod předzpracování. Hlavního přínosu pro úspěšnost klasifikace je dosaženo implementací tzv. kontextového (angl. Context aware) klasifikátoru, který je pak použit pro zbytek práce. V závěru je vylepšený algoritmus porovnán s výchozím algoritmem a s dalšími vybranými algoritmy. Dále je pak ověřena jeho úspěšnost na datech, které prošly simulovaným telefonním kanálem a na datech stejně upravených, které byly navíc zašumělé bílým šumem.

Klíčová slova

uzávěr hlasivek, hlasivkový puls, detekce, klasifikace, extreme gradient boosting, python, konvoluční neuronová síť, kontextový klasifikátor

Obsah

1 Úvod

S příchodem dialogových systémů, jako Kortana, Alexa či Siri do segmentu chytrých telefonů, hodinek a domácích asistentů, vyvstaly nové výzvy v oblastech komunikace s koncovým uživatelem. V zásadě se dají rozdělit do dvou oblastí, a to na porozumění lidské řeči a na syntézu umělé řeči. Při tvorbě umělé řeči je stále hojně využíváno metod konkatenační syntézy, kvalita takto syntetizované řeči velmi úzce závisí na přesné znalosti hlasivkových pulsů v řečovém signálu. Lepší kvalita v této oblasti znamená lepší uživatelský zážitek a tím i větší zisky výrobce. Samostatnou oblastí je pak identifikace řečníka a s tím spjaté zabezpečení takových systémů i zde je přesná znalost hlasivkových pulsů důležitou informací.

Předkládaná diplomová práce se zabývá detekcí hlasivkových pulsů v řečovém signálu a klade si za cíl vylepšení úspěšnosti klasifikace výchozího algoritmu. K tomuto účelu jsou použity metody strojového učení, konkrétně klasifikační algoritmus Gradient boosting, implementovaný v knihovně XGBoost. Mezi hlavní využité programovací prostředky se řadí programovací jazyk Python, program Octave a podpůrné programy Make a Git.

Úvodem se tato práce zabývá nastíněním problematiky detekce hlasivkových pulsů v řečovém signálu, jejich tvorby v hlasovém ústrojí člověka, zpracováním pomocí počítače a jejich významem pro další metody z oboru zpracování přirozeného jazyka umělou inteligencí, především pak v konkatenační syntéze řeči nebo v úlohách rozpoznávání řečníka. Poslední kapitola teoretického úvodu je věnována přehledu současně dostupných algoritmů pro detekci hlasivkových pulsů. V této práci je jako výchozí stav zvolen algoritmus vyvíjený na Katedře kybernetiky a je zde kladeno za cíl další zlepšení tohoto algoritmu. Vylepšení úspěšnosti klasifikace je docíleno nalezením dalších nových příznaků k výchozí množině příznaků. Ověřeno je také použití dalších způsobů předzpracování.

Následně jsou na rozšířenou množinu příznaků použity algoritmy pro výběr příznaků, které rozšířenou množinu zredukují na ty příznaky, které mají nejvyšší vypovídající hodnotou, bez větší ztráty úspěšnosti klasifikace algoritmu. Poté je na nejlepší podmnožině příznaků provedeno nalezení nových hodnot hyperparametrů klasifikátoru. V dalším úkolu je tento nejlepší nový algoritmus porovnán se způsobem předzpracování pomocí konvoluční neuronové sítě, natrénované na stejných datech, čili na stejné sadě promluv.

Oba porovnávané algoritmy jsou následně použity v dalším kroku pro tzv. kontextový klasifikátor, kde fungují jako předklasifikátor (použity jsou

separátně, buď jeden, nebo druhý), který určí pravděpodobnost hlasivkového pulsu. K této informaci od prvního klasifikátoru jsou volitelně přidány všechny původně vypočtené příznaky a také kontext pravděpodobností okolních kandidátů. Na tomto datasetu je poté natrénován další XGBoost klasifikátor a dohromady tvoří jeden kontextový klasifikátor.

V závěru této práce je celý vylepšený algoritmus porovnán s původním algoritmem a je ověřen na datech, která jsou pozměněna simulovanou telefonní linkou, a na datech stejně změněných s přidaným bílým šumem.

2 Problematika automatické detekce hlasivkových pulsů v řečovém signálu

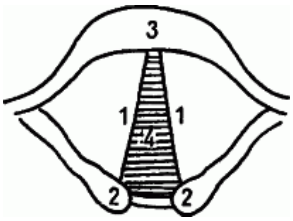
2.1 Způsob vytváření řeči

Tato kapitola si klade za cíl seznámit čtenáře s problematikou fyziologické tvorby hlasivkových pulsů a nastínit fyziologické pozadí a význam dále počítaných příznaků. Je zde čerpáno z prací Mluvíme s počítačem česky [49] a dále z článku [30], resp. z webové stránky [29].

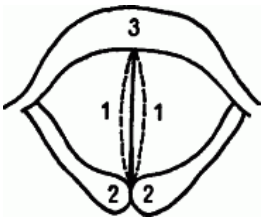
Na vytváření lidské řeči se podílí celá řada orgánů a svalů v těle, konkrétně v jeho hrudní, krční a lebeční části. Souhrnně nesou pojmenování hlasový trakt a lze je rozdělit na tři základní ústrojí: artikulační, hlasové a dechové.

Dechové ústrojí slouží v první řadě k dýchání, jakožto základní životní funkci, jeho druhořadá funkce je pak tvorba řeči, kde slouží jako zdroj energie. Uloženo je v hrudním koši. Postup vytváření řeči v dýchací soustavě lze zjednodušeně fyziologicky popsat následovně. Nejprve je do dechového ústrojí, především plic, nabrán vzduch, pomocí stahu bráničního svalu a uvolnění svalů hrudního koše. Následně je takto nahromaděný vzduch z plic uvolněn, čili dojde k uvolnění bráničního svalu a ke stahům svalů hrudního koše. Takto vzniklý vzduchový proud je poté veden průdušnicí do hlasového ústrojí.

Hlasové ústrojí najdeme v hrtanu a obsahuje nejdůležitější část pro tvorbu řeči, hlasivky, znázorněné na obrázku 2.1. Jedná se o dvě slizniční řasy, viz obrázek 2.1 číslo 1, umístěné příčně v nejužší části hrtanu, které jsou napojené na jedné straně na hlasivkové chrupavky, na obrázku 2.1 číslo 2, a na druhé straně na chrupavku příčnou, viz obrázek 2.1 číslo 3. Mezi sebou vytváří trojúhelníkovou štěrbinu, viz obrázek 2.1 číslo 4, kudy v případě, že člověk nemluví či generuje neznělou řeč, prochází vzduch volně. V případě, že člověk začne generovat znělou řeč, hlasivky se stáhnou k sobě a zúží onu štěrbinu. Pod tlakem vydechovaného vzduchu začnou hlasivky pravidelně kmitat a generovat tak vzduchové pulsy. Tyto periodické pulsy jsou označovány za základní hlasivkový tón a jeho frekvence potom za frekvenci základního hlasivkového tónu.



(a) Otevřená pozice hlasivek.



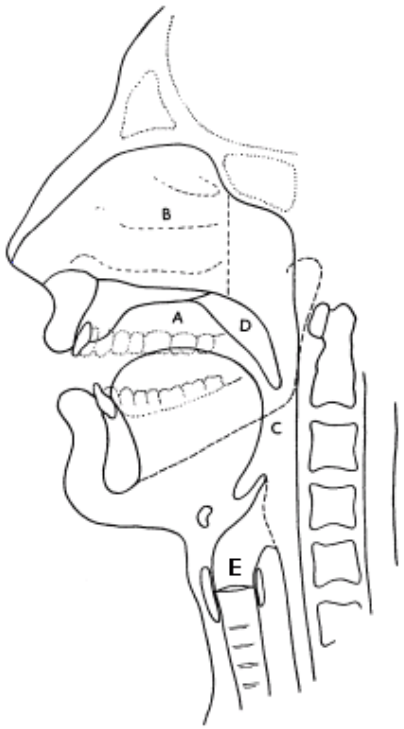
(b) Uzavřená pozice hlasivek při fonaci.

Obrázek 2.1: Schéma otevřené a uzavřené pozice hlasivek, převzato z [29].

Takto upravený proud vzduchu poté postupuje do artikulačního ústrojí, viz obrázek 2.2. To je uloženo nad hrtanem a skládá se z dutiny hrdelní, nosní a ústní, dohromady zvané nadhrtanové dutiny, a z artikulačních orgánů, z nichž nejdůležitější je jazyk. Ve výše zmiňovaných dutinách je zvuk dále výrazně modifikován.

Jednou takovou skupinou změn je tvorba tónové struktury řeči. K té dochází při průchodu periodického proudu vzduchu z hlasivek přes dutinu hrdelní a ústní, kdy je změněno rozložení akustické energie signálu. V takovém signálu můžeme pozorovat zajímavý jev, a sice koncentraci akustické energie okolo určitých frekvencí. Takové frekvence se nazývají **formantové frekvence** a oblasti zesílení akustické energie potom **formanty**. Značeny jsou pak od nejnižší frekvence jako $F1, F2, F3$ a dále. Pokud je do tvorby řeči zapojena i dutina nosní, může docházet k tvorbě takzvaných **antiformantů**, ty se značí $A1, A2, A3$ a dále, které naopak na určitých frekvencích akustickou energii potlačují.

Dutina hrdelní, viz písmeno C na obrázku 2.2, přímo navazuje na hlasivky, písmeno E na obrázku 2.2, a končí zhruba v místě styku jazyka a měkkého patra při artikulaci. Při fonaci je její hlavní úlohou tvořit rezonanční prostor. Ten může být různě objemný a je regulován stahy krčních a hrdelních svalů.



Obrázek 2.2: Schéma artikulačního ústrojí, převzato z [29].

Dutina nosní, na obrázku 2.2 písmeno B, je využívána jako rezonanční prostor jen u podmnožiny fonémů, jako například [m] nebo [n]. U většiny ostatních fonémů je průchodu vzduchu do této dutiny zabráněno přitisknutím vela, čili měkkého patra, viz obrázek 2.2 písmeno D, na zadní stěnu dutinu ústní. To je při dýchání volně spuštěno a vzduch může procházet nosem do plic a zpět.

Dutina ústní jako taková, znázorněna na obrázku 2.2 písmenem A, navazuje na dutinu hrdelní a lze ji počítat mezi pasivní artikulační orgány. Slouží především jako rezonanční prostor, opět s proměnlivým objemem, který je měněn především jazykem a rty. Spolu se změnami dutiny hrdelní se uplatňuje při tvorbě souhlásek, čili vytváří šumovou složku řeči. Například při tvoření souhlásky [s] vytvoří jazyk v dutině ústní hráz a výsledkem toho je pak výška šumu charakteristická právě pro tuto souhlásku. Dále pak dutina ústní obsahuje velmi důležité artikulační orgány, jako jazyk a rty.

Jazyk je ze všech aktivních artikulačních orgánů nejdůležitější a nejsložitější. Je tvořen především příčně pruhovanou svalovinou a je plně ovladatelný myslí při mluvení.

2.2 Zpracování řečového signálu v počítači

V této podkapitole bude přehledově vyložen proces převodu spojitého analogového signálu, jakožto výstupu artikulačního ústrojí, na jeho binární reprezentaci v počítačové paměti.

2.2.1 Zařízení pro záznam hlasu

V této podkapitole bude čerpáno, kromě jiného, z knihy Abeceda nf techniky [62]. Mezi základní zařízení pro záznam hlasu patří bezpochyby počítač či chytrý telefon, ty ale pracují pouze s diskrétními hodnotami. V prvé řadě je potřeba akustický signál určitým způsobem převést na signál elektrický, se kterým se těmto zařízením lépe pracuje. To je hlavní úloha mikrofonu. Jeho funkci v počítači, či chytrém telefonu lze zjednodušeně přiblížit následovně. Mikrofon je elektroakustický převodník, který přeměňuje akustickou energii, kterou nese mluvená řeč, ale například také šum prostředí, na energii elektrickou. Elektrické signály jsou vhodnější pro zpracování v počítači, nicméně ještě je potřeba převést spojitý elektrický signál na diskrétní posloupnost vzorků. Tento proces má na starosti A/D převodník, v počítači obsažený ve zvukové kartě [10].

2.2.2 Zvuková karta v PC a uložení nahraného zvukového signálu

V této sekci bude čerpáno z oficiální dokumentace pro programování hardware zvukové karty Sound Blaster [10]. Hlavními komponentami odpovědnými za převod analogového signálu na diskrétní a jeho následné uložení v počítači jsou zvuková karta, sběrnice, procesor a paměť resp. pevný disk počítače.

Rodina zvukových karet Sound Blaster patří v současné době k nejpoužívanějším zvukovým kartám. Jedná se dnes již o běžnou součást osobních počítačů a notebooků. Její hlavní činností je převod analogového signálu, například z mikrofonu, na diskrétní a opět diskrétní na analogový, například pro přehrávání hudby ve sluchátkách.

Pro převod z analogového na diskrétní signál je analogový signál nejprve vzorkován s dostatečnou vzorkovací frekvencí, která je minimálně dvojnásobek nejvyšší frekvence obsažené v signálu. To je realizováno odečtením hodnoty napětí v daných časových krocích. Následně je nutné provést kvantizaci naměřených hodnot napětí, jelikož počítače pracují jen s omezenou přesností čísel. To je realizováno pomocí kvantizační tabulky, resp. rozdě-

lením prostoru měření do kvantizačních úrovní. Reálné hodnotě je podle příslušnosti do kvantizační úrovně přiřazena daná hodnota. Odchylky od reálné hodnoty označujeme jako kvantizační šum. Čím hrubší je kvantizační mřížka, tím více šumu zaneseme do diskretizovaného signálu. Takto diskretizované hodnoty signálu jsou potom odesílány sběrnicí k dalšímu zpracování.

2.2.3 Formáty uložení audio dat, formát WAV

Po vzorkování signálu a přenesení sběrnicí, v našem případě karty Sound Blaster pomocí PCI (angl. Peripheral Component Interconnect) [10], případně PCIe (angl. PCI-express), jsou zvuková data uložena v paměti počítače. Pro jejich uložení je standardizovaná řada formátů. Audio formáty můžeme rozdělit do tří kategorií, a sice na bezkompresní, který bude dále popsán, kompresní bezztrátové, například FLAC (angl. Free Lossless Audio Codec) [8], a kompresní ztrátové, například MP3 (angl. MPEG-2 Audio Layer III) [50]. V této práci bylo nejvhodnější využití bezkompresního formátu. Při použití kompresního ztrátového formátu by došlo ke ztrátě potencionálně potřebné informace a použití kompresního bezztrátového formátu by přidalo do pracovního postupu zbytečný krok dekomprese.

Z bezkompresních formátů byl použit formát WAV (angl. Waveform Audio File Format) [20]. Jedná se o aplikaci formátu RIFF (angl. Resource Interchange File Format) [53]. Skládá se ze dvou částí - hlavičky a dat. Data jsou obvykle v 16bitové reprezentaci.

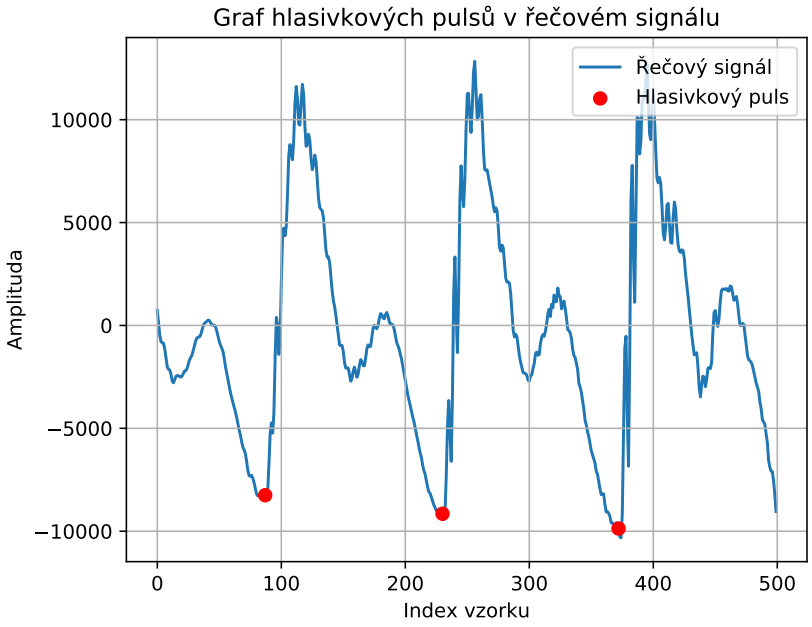
2.3 Analýza řečového signálu uloženého v počítači

V této diplomové práci byl jako hlavní technologie, pro zpracování dat a trénování klasifikátoru, zvolen programovací jazyk **Python** verze 3.6 [57]. Mezi jeho přední použité balíčky se řadí **numpy** [47], **scipy** [58], **pandas** [61], **scikit-learn** [48], **XGBoost** [6] a v neposlední řadě **librosa** [41]. Dále byl použit program a programovací jazyk **Octave** verze 5 a vyšší [13] s balíčky **signal** a **io**, který byl použit pro některé algoritmy předzpracování hlasového signálu, viz sekci 4.2. Jako operační systém pro provádění experimentů a běhu předzpracování byl zvolen **OS Fedora** verze 31 [52] s otevřeným zdrojovým kódem (angl. Open Source). V této práci bylo také využito výpočetních center **CESNET** (LM2015042), financovaných z programu MŠMT Projekty velkých infrastruktur pro VaVaI. V celé práci je brán záměrný zřetel na nástroje s otevřeným zdrojovým kódem a zároveň na jejich

multiplatformitu. Následující kapitola se zabývá popisem a analýzou hlasových signálů a hlasivkových pulsů s využitím představených technologií.

2.3.1 Hlasivkové pulsy v řečovém signálu

Po zpracování a uložení řečového signálu v počítačové paměti, může přijít na řadu analýza takového signálu. Na následujícím obrázku 2.3 je zobrazen časový průběh znělého zvukového signálu, to jest signálu, kdy hlasivky kmitají se základní hlasivkovou frekvencí. Jejich jednotlivé uzávěry (angl. Pitch Marks) jsou zvýrazněny červenými tečkami. Jejich rozdíl v časové oblasti je základní hlasivková perioda a její převrácená hodnota potom zmíněná základní hlasivková frekvence. Detekcí právě těchto červeně vyznačených hlasivkových uzávěrů se tato práce zabývá.

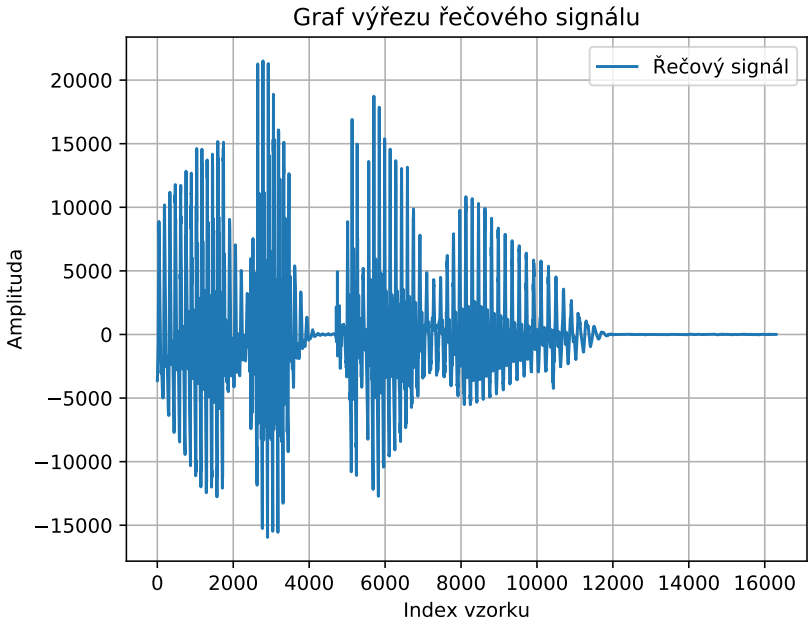


Obrázek 2.3: Pozice hlasivkových uzávěrů v řečovém signálu.

2.3.2 Frekvenční a časová analýza řečového signálu

V této podsekci bude čerpáno z knih [19] a [43] a z oficiální dokumentace balíčků scipy [58]. Grafy pro tuto práci byly vygenerovány balíčkem Matplotlib [18]. Pro extrakci příznaků (angl. Feature Extraction) byly využity především dvě oblasti reprezentace signálu a sice oblast časová a oblast frekvenční. V časové oblasti, ve které je zvukový signál ve formátu WAV

uchováván, je možné využít například míry v eukleidovském prostoru. Nahraný zvuk lze reprezentovat, jako závislost kvantizované amplitudy signálu na čase, resp. pozici amplitudy v poli. Takový časový průběh signálu je vyobrazen na obrázku 2.4



Obrázek 2.4: Časový průběh vybraného řečového signálu.

V této reprezentaci lze měřit vzdálenost dvou bodů v prostoru amplitudy a času, resp. pozici v poli. Takto lze získat například amplitudu hlasivkového pulsu, vzdálenost dvou hlasivkových pulsů nebo jejich šířku.

V časové oblasti dále můžeme využít například korelační analýzy. Díky ní lze vypočítat například **Pearsonův korelační koeficient** mezi potenciálními hlasivkovými pulsy. Tento koeficient je využit jako jeden z příznaků výchozího algoritmu. K výpočtu lze využít vzorec ve tvaru

$$r_{ij} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}. \tag{2.1}$$

.

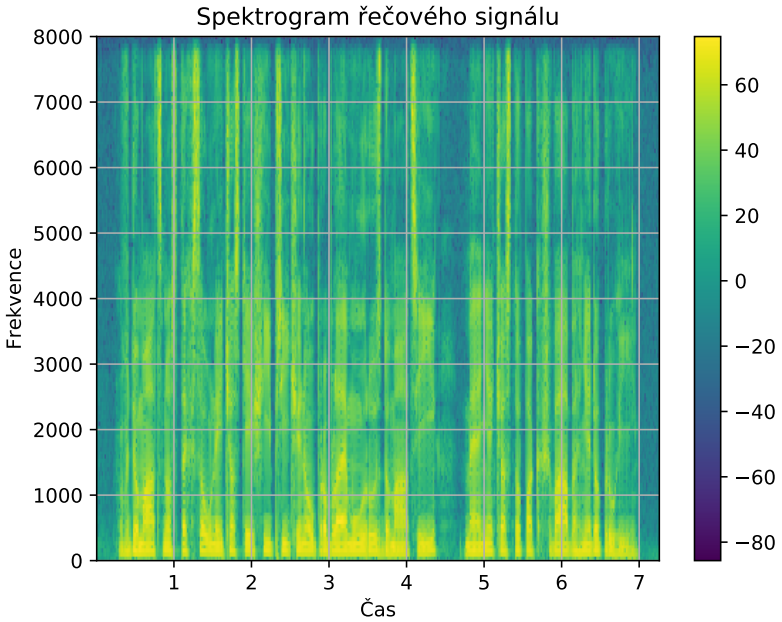
Další významnou oblastí pro extrakci příznaků, je oblast frekvenční. Do té převedeme signál pomocí **Fourierovy transformace (FT)**, která signál vyjádří v bázi prostoru složené ze sinů a cosinů. Zde lze pak sledovat další zajímavé vlastnosti signálu v závislostech, jako je amplitudové frekvenční spektrum, spektrogram, který znázorňuje závislost frekvenci na čase a amplitudě, nebo lze spočítat tzv. melovské kepstrální koeficienty, které Fourierovu

transformaci využívají. Fourierovu transformaci však nemůžeme použít na obecný signál, jelikož nemůžeme zaručit stacionaritu takového signálu. Lze však použít algoritmus **krátkodobé Fourierovy transformace (STFT)**, kterému postačuje stacionarita po krátkých časových intervalech, což v řeči zaručit můžeme, jelikož artikulační ústrojí nedokáže měnit nekonečně rychle svoje nastavení. Algoritmus krátkodobé Fourierovy transformace postupuje tak, že pro dané klouzavé okénko vypočítává **diskrétní Fourierovu transformaci (DFT)** algoritmem **rychlé Fourierovy transformace (FFT)**, DFT má následující tvar, kde $\{x_0, ..., x_{N-1}\} \subset \mathbb{C}$ jsou vzorky signálu,

$$\hat{X}[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-i2\pi kn}{N}}, k = 0, ..., N-1. \tag{2.2}$$

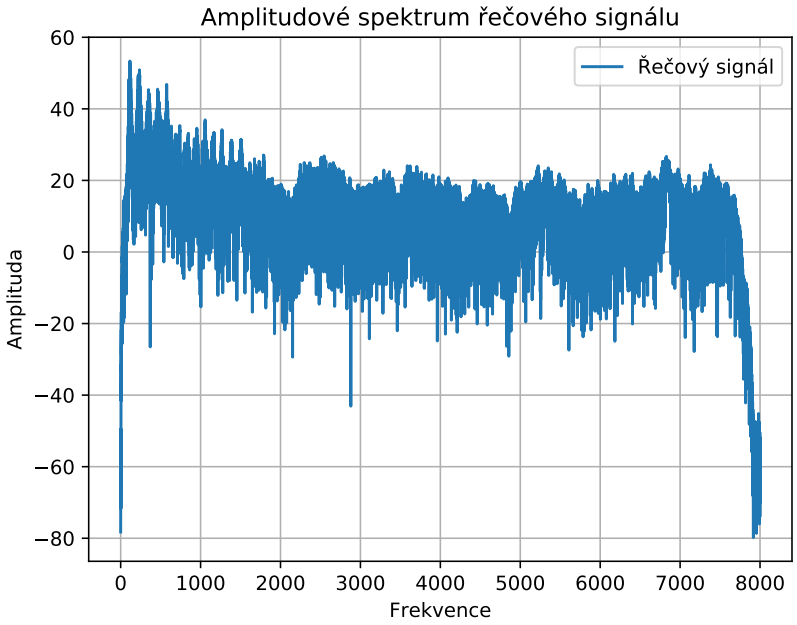
Algoritmus rychlé Fourierovy transformace výrazně zrychluje výpočet Fourierovy transformace, konkrétně se ze složitosti 2^N dostáváme na $Nlog_2N$, potřebuje však sudý počet vzorků.

Na obrázku 2.5 je znázorněn spektrogram signálu znázorněného na předchozím obrázku 2.4.



Obrázek 2.5: Spektrogram vybraného signálu.

A na dalším obrázku 2.6 je potom znázorněno amplitudové frekvenční spektrum signálu.



Obrázek 2.6: Amplitudové spektrum úseku vybraného signálu.

2.4 Klasifikační algoritmus XGBoost

XGBoost, celým názvem Extreme Gradient Boosting, kde označení **Gradient Boosting** pochází z původního článku [14], je klasifikační algoritmus učení s učitelem (angl. Supervised Learning). Jedná se o knihovnu s otevřeným zdrojovým kódem, která implementuje algoritmus Gradient Boosting pro jazyk Python. Základ Gradient Boosting algoritmu tvoří rozhodovací stromy resp. soubory rozhodovacích stromů zvané CART, čili klasifikační a regresní stromy (angl. Classification and Regression Trees) pevně dané hloubky.

Algoritmus postupuje iterativně a kombinuje jednotlivé stromy, tzv. **slabé učence** (angl. Weak Learner), do jednoho algoritmu zvaného **silný učenec** (angl. Strong Learner). Gradient v názvu odkazuje na použití algoritmu Gradient descent během výpočtů nových parametrů. Ve zkratce gradient boosting algoritmus pro daný dataset a informaci od učitele natrénuje první strom. Poté dojde ke klasifikaci a určení reziduí, tzn. chyb od reálných hodnot. Tato rezidua jsou následně aktualizována pomocí algoritmu gradient descent dané hodnotící funkce. Na těchto nově vypočtených reziduích je natrénován další strom a algoritmus pokračuje ve stejném duchu dále, dokud nejsou natrénovány všechny stromy specifikované v úvodu algoritmu. Algo-

ritmus konverguje lokálně, z tohoto důvodu je potřeba provést více běhů a výsledky statisticky vyhodnotit.

Pro různé typy klasifikátorů, které jsou v této práci použity pro experimenty bylo zavedeno jednotné značení. To je vždy vysvětleno v daném popisu experimentu a pro přehlednost je uvedeno na konci práce v příloze A.

2.5 Způsoby vyhodnocení úspěšnosti klasifikátoru

Po natrénování klasifikátoru je potřeba určit jeho úspěšnost. To se provádí buď na testovacích datech, která jsou stranou celého trénovacího procesu, nebo na složce (angl. Fold) dat, která klasifikátor ještě neviděl. Metriky použité v této práci pro vyhodnocení klasifikátoru lze rozdělit na dvě kategorie, a sice na metriky klasifikační a metriky specializované na porovnávaní algoritmů pro detekci hlasivkových pulsů.

Obě kategorie metrik používají označení kandidát, popřípadě kandidát na hlasivkový puls, jedná se o potencionální hlasivkový puls, který ovšem ještě nebyl klasifikován. Po klasifikaci lze kandidáty rozdělit do čtyř kategorií, na správně označené hlasivkové pulsy (angl. True Positive, **TP**), na správně označené nehlasivkové pulsy (angl. True Negative, **TN**), dále na špatně označené nehlasivkové pulsy (angl. False Positive, **FP**) a v poslední řadě na špatně označené hlasivkové pulsy (angl. False Negative, **FN**). Tyto kategorie jsou přehledně vypsány v následující tabulce 2.1.

Tabulka 2.1: Tabulka kategorií klasifikovaných kandidátů.

	Hlasivkový puls	**Nehlasivkový puls**
Klasifikovaný hlasivkový puls	TP	FP
Klasifikovaný nehlasivkový puls	FN	TN

V této práci jsou nejprve určeni z řečového signálu všichni přípustní kandidáti, kteří jsou následně klasifikováni.

2.5.1 Klasifikační metriky

Pro popis těchto metrik bylo čerpáno z oficiální dokumentace balíčku **scikit-learn** [5].

Preciznost (angl. Precision)

Jedná se o poměr správně klasifikovaných kandidátů na hlasivkový puls, ku součtu správně klasifikovaných kandidátů na hlasivkový puls a špatně klasifikovaných kandidátů na hlasivkový puls. Tato metrika nabývá hodnot mezi 0 a 1, přičemž nejlepší je 1 a lze ji vyjádřit, jako

$$P = \frac{TP}{(TP + FP)}. \tag{2.3}$$

Průměrná přesnost (angl. Average Precision)

Počítá průměrnou přesnost z predikcí klasifikátoru podle následujícího vzorce

$$AP = \sum_{n} (R_n - R_{n-1}) P_n, \tag{2.4}$$

kde R_n značí recall a P_n značí preciznost. I zde je nejlepší vásledek 1 z intervalu 0 až 1.

Recall

Zde se jedná o poměr správně klasifikovaných kandidátů na hlasivkový puls, ku součtu správně klasifikovaných kandidátů na hlasivkový puls a špatně klasifikovaných kandidátů jako nehlasivkové pulsy. Lze vypočítat následovně

$$R = \frac{TP}{(TP + FN)}, \tag{2.5}$$

přičemž nejlepší vásledek je 1 z intervalu 0 až 1.

F1

Míra F1 může být interpretována jako vážený průměr měr preciznost a recall, s rozsahem 0 až 1, kde 1 je nejlepší. Lze vypočítat takto

$$F1 = \frac{2 \cdot (P \cdot R)}{(P + R)}. \tag{2.6}$$

Přesnost (angl. Accuracy)

Tato míra počítá procentuální shodu predikcí klasifikátoru s informací od učitele. Lze spočítat, jako

$$AC = \frac{1}{N}\sum_{i=0}^{N-1} 1(\hat{y}_i = y_i), \tag{2.7}$$

kde $\hat{y}_i$ je informace od učitele, y_i je výstup klasifikátoru pro kandidát a s rozsahem 0 až 1, kde 1 je nejlepší.

Balancovaná přesnost (angl. Balanced Accuracy)

Je obdoba předchozího a je definována jako průměr míry recall, která je získána pro každou třídu dat. Lze ji vypočítat následujícím předpisem

$$BAC = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}). \tag{2.8}$$

Opět platí rozsah 0 až 1, kde 1 je nejlepší.

Plocha pod ROC křivkou (angl. ROC AUC)

Zde je počítána oblast pod křivkou operační charakteristiky přijímače (angl. Reciever Operating Characteristic Area Under Curve, ROC AUC). Ta je získána, jako závislost poměru TPR (angl. True Positive Rate, **TPR**) a poměru FPR (angl. False Positive Rate, **FPR**), opět s rozsahem 0 až 1, kde 1 je nejlepší. Tyto poměry jsou počítány podle následujících vzorců

$$TPR = \frac{TP}{TP+FN} \tag{2.9}$$

a

$$FPR = \frac{FP}{FP+TN}. \tag{2.10}$$

Briérovo skóre (angl. Brier score)

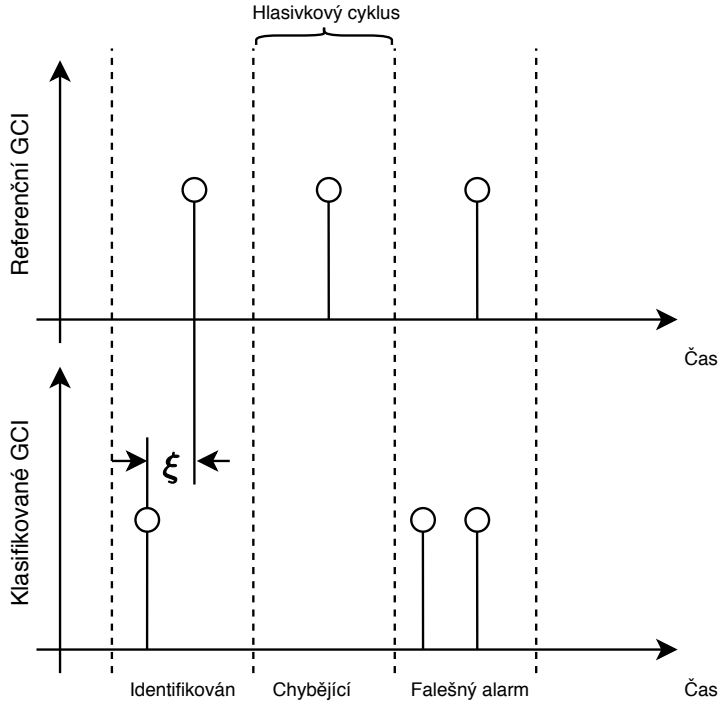
Měří střední kvadratickou chybu mezi predikovanou pravděpodobností pro kandidáta, že jde o hlasivkový puls a informací od učitele. Čím nižší je Brierovo skóre pro sadu předpovědí, tím lépe je klasifikátor nastaven. Brierovo skóre vždy nabývá hodnoty mezi 0 a 1 a lze vypočítat, jako

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2, \tag{2.11}$$

kde f_t je pravděpodobnost kandidáta na hlasivkový puls a o_t je binární informace od učitele.

2.5.2 Specializované metriky

V této podkapitole bude čerpáno z článků [35, 45]. Na následujícím obrázku 2.7 jsou názorně ukázány dále počítané míry. Je zde pro zkrácení použita zkratka GCI (angl. Glottal Closure Instant) a znamená hlasivkový puls.



Obrázek 2.7: Znázornění jednotlivých specializovaných metrik (IDR, MR, FAR a IDA), převzato z [12].

Míra identifikace (angl. Identification Rate, IDR)

Jedná se o procentuální poměr hlasivkových cyklů s právě jedním klasifikovaným hlasivkovým pulsem ku všem hlasivkovým cyklům. Hlasivkovým cyklem je zde myšlena základní hlasivková perioda, tedy čas od uzávěru hlasivek přes excitaci, až do dalšího uzávěru. Na obrázku 2.7 je znázorněna v intervalu „Identifikován“.

Míra vynechání (angl. Miss Rate, MR)

Zde jde o procentuální vyjádření počtu hlasivkových cyklů, pro které nebyl žádný kandidát klasifikován, ku všem hlasivkovým cyklům. Na obrázku 2.7 je znázorněna intervalem „Chybějící“.

Míra falešných alarmů (angl. False Alarm Rate, FAR)

Naopak tato míra je procentuální vyjádření hlasivkových cyklů, pro které byl více než jeden kandidát klasifikován, jako hlasivkový puls, ku všem hlasivkovým cyklům. Na obrázku 2.7 je znázorněna v intervalu „Falešný alarm“. Nutno podotknout, že $IDR + MR + FAR = 100\ \%$.

Identifikační přesnost (angl. Identification Accuracy, IDA)

Tato míra vyjadřuje v milisekundách střední hodnotu směrodatné odchylky chyby klasifikace hlasivkových pulsů. Na obrázku 2.7 je znázorněna v intervalu „Identifikován“ řeckým písmenem ξ.

A25

Jedná se o procento správně klasifikovaných kandidátů v povoleném intervalu chyby od $-25ms$ do $+25ms$ ze všech hlasivkových kandidátů. Jedná se o přísnější obdobu IDR.

E10

Je míra definovaná, jako

$$E10 = \frac{N_R - N_{\zeta > 0.1T_0} - N_M - N_{FA}}{N_R}, \tag{2.12}$$

kde N_R reprezentuje počet referenčních hlasivkových pulsů, N_M je počet neklasifikovaných hlasivkových pulsů (viz MR), N_{FA} je počet špatně klasifikovaných hlasivkových pulsů, podobně jako FAR, a $N_{\zeta > 0,1T_0}$ je počet hlasivkových pulsů s identifikační chybou ζ větší něž 10 % lokální hlasivkové periody T_0 [35].

2.5.3 Statistické porovnání klasifikátorů

McNemarův test

V této podsekci je čerpáno z článku [42] a z oficiální dokumentace balíčku **mlxtend** [51]. Jedná se o statistický test pro párové uspořádání experimentu, který sleduje kvalitativní výskyt náhodné veličiny X na stejném výběrovém souboru dvakrát po sobě. V kontextu strojového učení lze tento test použít pro porovnání prediktivní přesnosti dvou modelů. Tento test je založen na 2x2 kontingenční tabulce sestrojenou pro oba modely a její definice je v následující tabulce 2.2.

Tabulka 2.2: Kontingenční tabulka McNemarova testu.

	Správná klasifikace model 1	**Špatná klasifikace model 1**
Správná klasifikace model 2	s	b
Špatná klasifikace model 2	c	d

McNemarova statistika je pak počítána z následujícího vzorce, jedná se o chí kvadrát rozdělení s jedním stupněm volnosti,

$$\chi^2 = \frac{(b-c)^2}{(b+c)}. \tag{2.13}$$

Lze také vypočítat p hodnotu. Na základě těchto vypočtených charakteristik lze rozhodnout o přijetí či nepřijetí nulové hypotézy. Nulová hypotéza pro tento test je, že pravděpodobnost správné klasifikace je stejná u obou modelů. Alternativní hypotéza je, že se liší. V případě, že je p hodnota menší, než zvolený práh, zamítámé nulovou hypotézu.

Notched Box and Whiskers graf a statistická významnost

Pro porovnání výkonnosti dvou a více klasifikátorů lze použít i tzv. box plot se zářezy (angl. Notched box Plot). V případě, že se zářezy dvou boxů nepřekrývají lze prohlásit jejich mediány za statisticky významně rozdílné, výchozí hladina významnosti je opět $\alpha = 0{,}05$ [31].

2.6 Význam hlasivkových pulsů pro syntézu řeči a další odvětví zpracování řečového signálu

V této podkapitole je čerpáno z článku [12]. Hlasivkové pulsy v řečovém signálu lze přesně změřit pomocí přístroje zvaného elektroglotograf (EGG) [17]. Signál z něj vystupující dává přesnou informaci o pozici hlasivkových pulsů. Zvyšující se zájem o metody zpracování řeči, využívající informaci o pozici hlasivkových pulsů (angl. Pitch Synchronous Methods),

s sebou přinesl také vetší potřebu jejich korektní detekce a to jak ze studiové, čili čisté řeči, tak z řeči obecné, či zašumělé [12].

Přesná znalost hlasivkových pulsů je stěžejní pro přesnost a kvalitu různých metod z oblasti zpracování řeči. Jednou z nich je například řečová syntéza nebo konkrétněji konkatenační syntéza, například metoda **TD-PSOLA**. V jejím případě má přesná detekce hlasivkových pulsů a z toho plynoucí přesné určení fundamentální frekvence přímý dopad na kvalitu syntetizované řeči [9], nebo například na kvalitu transformace řeči [33].

Další obdobnou oblastí, kde je přesné určení uzávěrů hlasivek kritické, je úloha rozpoznávání řečníka. V těchto systémech figurují například kontury základní hlasivkové frekvence, jako jeden z důležitých příznaků. K přesnému určení fundamentální hlasivkové frekvence je potřebná co nejpřesnější znalost hlasivkových pulsů [25, 32].

Určení pozice hlasivkových pulsů je důležité i pro oblast prozodických modifikací řeči [44], jako například změna výšky tónu, rychlost mluvení, hlasitost řeči a v perceptuální oblasti změna melodie a rytmu řeči. Takové modifikace jsou potřebné i pro další metody zpracování řeči.

2.7 Přehled dostupných algoritmů detekce hlasivkových pulsů v řečovém signálu

V této podkapitole bude představen stav aktuálního vývoje (angl. State of the Art), čili přehled aktuálně dostupných algoritmů pro detekci hlasivkových pulsů. Jako základní podklad byly využity články [12, 21].

2.7.1 Zero frequency filtering (ZFF) algoritmus

Metoda ZFF [27] používá techniky filtrování řečového signálu přes dva do kaskády zapojené ideální rezonátory s nulovou frekvencí. Základním principem je fakt, že impulsní excitace hlasivek, čili hlasivkový puls, má efekt přes všechny frekvence spektra. Výstupní signál je označován jako „zero-frequency filtered“ a jako kandidáty na uzávěry hlasivek jsou vybírány přechody přes nulu ze záporné do pozitivní oblasti amplitud signálu.

2.7.2 SEDREAMS algoritmus

Metoda SEDREAMS [11, 12] má v této práci větší význam, než ostatní metody zmíněné v této podkapitole, jelikož její výstupní signál, založený na střední hodnotě (angl. Mean-based Signal - MS), byl testován jako jeden ze

způsobů předzpracování signálu. Tato metoda spočívá ve využití jak excitačního, tak řečového signálu. Z nich je počítán tzv. signál založený na střední hodnotě (angl. Mean-based Signal, MS). Výpočet využívá střední hodnoty klouzajícího okénka velkosti rovné 1,75 krát průměrná fundamentální frekvence, která je odhadnutá z řečového signálu. První odhad uzávěru hlasivek je proveden z MS signálu a následně je zpřesněn využitím excitačního signálu.

2.7.3 SE-VQ algoritmus [22]

Jedná se o modifikaci SEDREAMS algoritmu. Kromě obdobného postupu jako v případě SEDREAMS, jsou zde aplikovány metody dynamického programování, pro vybrání optimálních kandidátů na hlasivkový puls, a následně je prováděno zpracování pro odstranění falešně pozitivních kandidátů.

2.7.4 Dynamic programming phase slope algorithm (DYPSA)

DYPSA algoritmus [46] využívá pouze excitační signál k detekci hlasivkových pulsů. Tato metoda nejprve určí průchody nulou fázové funkce vypočtené z lineárního predikčního (LP) reziduálního signálu, aby získala kandidáty na hlasivkové pulsy. Poté jsou pomocí fázové funkce určeni dodateční kandidáti, které první krok neodhalil. Následně jsou využity metody dynamického programování pro vyřazení falešně positivních kandidátů.

2.7.5 Yet Another GCI Algorithm (YAGA)

V překladu další algoritmus pro detekci hlasivkových pulsů [56]. Tato metoda sdružuje více přístupů již zmíněných algoritmů, například vlnkovou (angl. Wavelet) analýzu, tzv. skupinově zpožděnou funkci (angl. Group Delay Function) a M-nejlepší dynamické programování. Kvůli zdůraznění nespojitostí v řečové signálu je spočítán vícerozměrný kartézský součin stacionární vlnkové transformace. Z výsledného signálu jsou detekovány nespojitosti pomocí průchodů nulou skupinově zpožděné funkce. Následně je postupováno obdobně jako v metodě DYPSA, čili je využito metod dynamického programování.

2.7.6 Microcanonical multi-scale formalism (MFF)

Algoritmus MFF [24] využívá přímý rozvoj impulsní nespojitosti v signálu pomocí nelineárního formalismu. Tato metoda je postavená na tzv. microcanonical multi-scale formalismu a závisí na přesném odhadu vícerozměrného parametru zvaného exponent singularity.

2.7.7 Reaper algoritmus

Reaper algoritmus [55] nejprve odstraní filtrem typu horní propusť nízkofrekvenční šum. Následně je provedena extrakce příznaků pro jednotlivé kandidáty na hlasivkový puls, například normalizovaná kroskorelace. Kandidátům jsou potom určeni jejich sousedé v kontextu odpovídajícímu intervalu od minimální do maximální periody hlasivkového pulsu. V tomto celém kontextu jsou následně kandidáti ohodnoceni, jakožto hypotéza, že se jedná o hlasivkový puls. Z této kontextové mřížky je potom metodou dynamického programování nalezena optimální cesta, čili optimální hlasivkové pulsy.

2.7.8 Glottal closure/opening instant Estimation Forward-Backward Algorithm (GEFBA)

GEFBA [28] je algoritmus pro simultánní detekci řeči v signálu a následnou detekci uzavření hlasivek právě a jen z řečové části signálu. Odhad uzavření hlasivek je proveden na základě jednoduchého kritéria v časové oblasti. Jak již bylo řečeno, GEFBA je také detektor řeči v signálu, který je schopen detekce s vysokým rozlišením i u konců řečových segmentů. Na čistém řečovém signálu vykazuje lepší úspěšnost než výše zmíněný SEDREAMS, nebo YAGA.

2.7.9 Probabilistic source-filter model (PSFM)

PSFM [2] využívá principu pravděpodobnostního modelu tvorby řeči. Hlasivkové pulsy jsou modelovány Bernoulli Gaussovo distribucí (BG), která modeluje jejich pozici v signálu a sílu excitace. Pravděpodobnost, že kandidát na hlasivkový puls je skutečně hlasivkový puls je odhadnuta pomocí Gibbsova vzorkování. Následně jsou určeny jednotlivé uzavření hlasivek pomocí M-nejlepšího dynamického programování.

3 Výchozí algoritmus

V této kapitole bude podrobně představen výchozí algoritmus, vyvíjený na Katedře kybernetiky Západočeské univerzity v Plzni. Tento algoritmus slouží jako základ pro tuto práci, která si klade za cíl jeho rozšíření a vylepšení úspěšnosti klasifikace hlasivkových pulsů. Základem je tedy kód, který byl k této práci předaný, ten obsahuje skripty v jazyce Python pro předzpracování signálu, množinu promluv v různých jazycích a od různých řečníků ve formátu WAV, soubory s referenčními hlasivkovými pulsy a základní experiment s klasifikátorem XGBoost. Jako dokumentace slouží tři po sobě vydané články [37–39], ze kterých je v této kapitole čerpáno.

3.1 Použité metody předzpracování

Pro vytvoření množiny dat je připraven skript `make_utt_feats.py`, v něm jsou nejprve načteny soubory promluv a soubory s referenčními hlasivkovými pulsy v jednotlivých promluvách. Následně jsou jednotlivé zvukové soubory procházeny sekvenčně. Každý jednotlivý soubor je přečten a v případě potřeby převzorkován. Data jsou, buď již původně, nebo po převzorkování, vzorkována s frekvencí 16 kHz. Následně je pro každý načtený zvukový soubor vytvořena normalizovaná reprezentace s maximální amplitudou 30000, tím bylo docíleno převedení obecného řečového signálu na celočíselný datový typ s velikostí 16 bitů. Původní načtený signál je pak filtrován filtrem typu dolní propusť s nulovou fází a danými koeficienty uloženými v souboru `filtcoef800.npy` ve složce `Data`. Tím je odstraněn vysokofrekvenční šum. Vzhledem k faktu, že nahrávky jsou pořízené ve studiu, nebylo nutné filtrovat také nízkofrekvenční šum, například z elektrospotřebičů. Po filtraci je signál opět normalizován na celočíselný datový typ s velikostí 16 bitů. Výstupem předzpracování jsou tři signály normalizovaný, filtrovaný a originální, na kterých jsou následně počítány jednotlivé příznaky.

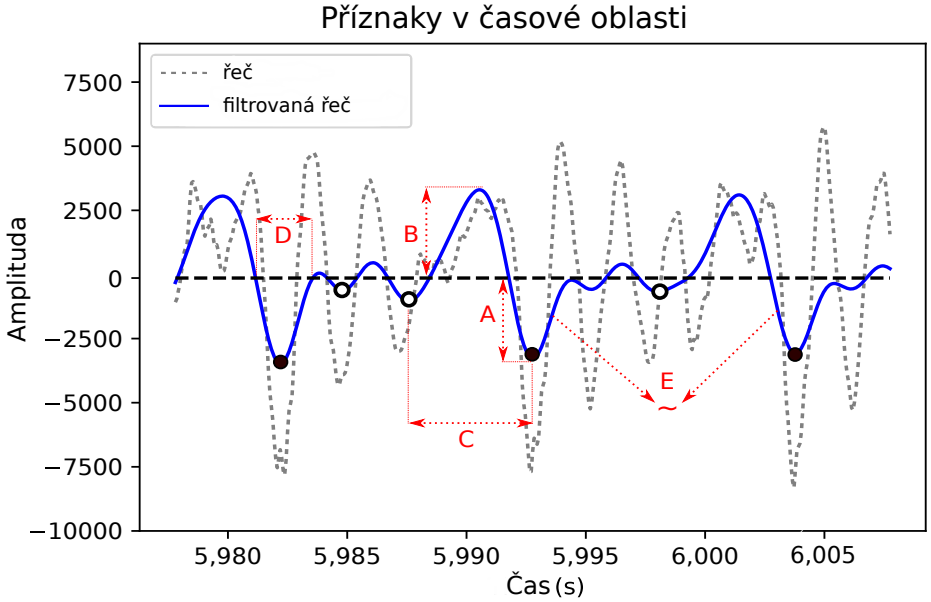
3.2 Příznaky pro strojové učení

V prvé řadě byl vytvořen prázdný dataset, který je instancí objektu `dataFrame` balíčku **pandas**, s názvem stejným, jako název zvukového souboru. Jako první sloupec je do datasetu přidán sloupec s názvem **negPeakIdx**, který obsahuje indexy negativních vrcholů v signálu, které byly pře-

dem detekovány metodou `detect_peaks` balíčku `wav_manip`[1]. Princip jejich detekce je takový, že jsou nejprve určeny průchody nulou signálu a mezi záporným a kladným průchodem je určen největší negativní vrchol. Negativní polarizace byla určena v návrhu algoritmu. V zásadě nemá polarizace vliv na výsledek, musí však být stejná pro všechny vrcholy, buď kladná, nebo záporná. Dále bylo třeba ohodnotit detekované kandidáty na hlasivkové pulsy informací od učitele. Ta je obsažena v souboru `PM` se jménem stejným, jako jméno zvukového souboru. Ten obsahuje informaci z EGG, avšak pouze interval času, kde se skutečný hlasivkový puls nachází. Je tedy potřeba použít metodu `gci2targets`, která ohodnotí každého kandidáta na hlasivkový puls na základně minimální vzdálenosti od informace v souboru `PM`. Následně jsou již sekvenčně počítány jednotlivé příznaky a postupně přidávány do datasetu pro jednotlivé zvukové soubory. Všechny datasety jednotlivých nahrávek jsou na konci sloučeny do jednoho datasetu s vynecháním informace o pozici kandidáta v nahrávce.

3.2.1 Příznaky v časové oblasti

Následující obrázkek 3.1 znázorňuje příznaky počítané v časové oblasti.



Obrázek 3.1: Příznaky v řečovém signálu.

[1]Tato knihovna je přístupná na přiloženém CD ve složce *Preprocessing/lib*.

Amplituda negativních vrcholů

První z počítaných příznaků jsou amplitudy negativních vrcholů v signálu s kontextem 3 vrcholy, toto číslo bylo určeno v [37] a znamená zaznamenání 3 předchozích a 3 následujících příznaků kandidátů. S tímto kontextem je počítáno i u dalších příznaků. Přidáno tedy bude 7 sloupců postupně **NegAmp-3** až **negAmp+3**. Jedná se o amplitudy kandidátů na hlasivkové pulsy počítané z filtrovaného signálu, znázorněny na obrázku 3.1 písmenem A.

Amplituda pozitivních vrcholů

Obdobně jsou počítány amplitudy pozitivních vrcholů opět s kontextem 3 vrcholy, tentokrát nesou sloupce označení **PosAmp-3** až **PosAmp+3**. Na obrázku 3.1 znázorněné písmenem B.

Časový rozdíl vrcholů

Tento příznak značí přepočtenou délku trvání vrcholu v čase a opět je počítán pro kontext 3 vrcholy, tedy jsou přidány sloupce **timeDif-3** až **timeDif+3**. Na obrázku 3.1 je znázorněn písmenem C.

Šířka vrcholů

Následuje výpočet šířky vrcholů v signálu. Je vytvořena instance třídy `FeatPeakWidth` a pro všechny indexy vrcholů je spočítán levý průběh nulou od indexu vrcholu a pravý průběh nulou od indexu vrcholu a následně je jejich rozdíl v indexech, čili šířka, normován a přidán, jako příznak do datasetu. Tento postup je opět rozšířen o kontext 3 ke každému vrcholu, to znamená, že v datasetu jsou sloupce **width-3** až **width+3** s tím, že **width0** je šířka aktuálně zkoumaného vrcholu. Na obrázku 3.1 je znázorněn písmenem D.

Korelace vrcholů

Korelací jednotlivých vrcholů se rozumí výpočet korelačního koeficientu, viz obrázek 2.3.2, daného vrcholu s danou polaritou a s daným počtem, v našem případě postupně s kontextem 3 vrcholy. V datasetu pak nalezneme sloupce **corr-3** až **corr+3**. Na obrázku 3.1 je tento příznak znázorněn písmenem E.

Poměr vrcholů

Zde je počítán poměr amplitudy daného vrcholu k maximální amplitudě, která se v signálu vyskytuje. Opět je přidán kontext 3 vrcholů a všechny jsou zapsány v datasetu do sloupců **negPeakRatio-3** až **negPeakRatio+3**.

Míra průchodů nulou

V tomto příznaku je počítána míra, tedy počet průchodů nulou v okolí daného vrcholu. Všude dále bude bráno okolí velikosti 10 ms. Při vzorkovací frekvenci 16 kHz to znamená okolí 160 vzorků, které je centrované okolo zkoumaného vrcholu. Tento příznak je v datasetu označen jako **zcr**.

3.2.2 Příznaky ve frekvenční oblasti

Logaritmus energie

Tento příznak je počítán opět pro dané okolí každého vrcholu v signálu. Pro každý index vrcholu je vybráno okolí vzorků a pro tento extrahovaný signál je pomocí funkce `mfcc` z balíčku `python_speech_features` [36] získán logaritmus krátkodobé energie signálu. Jelikož je ve funkci zapnuta volba `appendEnergy` na pravda, je místo nultého melovského kepstrálního koeficientu vrácen odhad logaritmu krátkodobé energie signálu. Okénková funkce je zde využívána typu Hamming. Po výpočtu jsou opět hodnoty zapsány do datasetu, tentokrát do sloupce **energy**.

Poměr harmonického signálu k šumu

Za další je počítán poměr harmonického signálu k šumu. K tomu je využita metoda `get_HNR` balíčku `Signal_Analysis`. Opět je tento příznak počítán pro dané okolí a jako práh ticha je zvoleno 10 % z maximální amplitudy v signálu. Následuje přidání vypočtených hodnot do datasetu, jako sloupce s názvem **hnr**.

Spektrální centroid

Metodu pro výpočet spektrálního centroidu nám nyní poskytne balíček **librosa**. Pro výpočet krátkodobé Fourierovy transformace bylo použito okénko velikosti 512 vzorků. Spektrální centroid je počítán pro dané okénko signálu, čili pro 160 vzorků. Spektrální centroid je míra, která udává, na jakých frekvencích se většina masy spektra signálu nachází. V datasetu je pak reprezentován jako sloupce **specCentroid**.

Spektrální šířka pásma

Tento příznak je opět počítán s pomocí balíčku **librosa**. Pro výpočet krátkodobé Fourierovy transformace bylo opět použito okénko velikosti 512 vzorků. A samotná spektrální šířka pásma je počítána pro dané okénko okolo vrcholu, 160 vzorků. Více o spektrální šířce pásma, viz článek [26]. Do datasetu je přidán jako sloupec **specBandwidth**.

Spektrální roll-off

Jedná se o další z příznaků, počítaných pomocí funkce z knihovny `librosa`. Tentokrát se jedná o frekvenci, která je počítána ze spektrogramu signálu, vybraného pro dané okolí vrcholu. Nejprve je vypočítán spektrogram a poté je určena frekvence, ve které je obsaženo 85 % energie signálu. Tato frekvence je poté zapsána pro daný vrchol do sloupce **specRollOff**.

MFCC

S výpočtem kepstrálních koeficientů se v této práci již pracovalo a to při výpočtu příznaku logaritmus energie okolí vrcholu. I zde je použita stejná funkce `mfcc` z balíčku `python_speech_features` a je vypočítáno 13 melovských kepstrálních koeficientů pro dané okolí vrcholu a pro danou velikost okénka rychlé Fourierovy transformace. Pro každý vrchol jsou pak přidány dané koeficienty do sloupců **mfcc0** až **mfcc12**.

Základní hlasivková frekvence

Pro odhad základní hlasivkové frekvence je zde využit algoritmus REAPER (angl. Robust Epoch and Pitch Estimator) z knihovny `pyreaper`, která obaluje tento algoritmus. Odhad základní hlasivkové frekvence je proveden na celém neupraveném signálu a následně jsou vybrány odhady odpovídajícím pozicím jednotlivých vrcholů. Tento příznak je do datasetu zařazen jako sloupec **f0**.

3.3 Klasifikátor

Po vygenerování datasetů pro jednotlivé promluvy, jež bylo představeno v kapitole výše, jsou všechny tyto dílčí datasety zkombinovány do jednoho, při vypuštění informace o indexu vrcholu. Poté přichází na řadu trénování vybraného klasifikátoru. Výběr nejvhodnějšího klasifikátoru byl proveden v článku [37]. Vybrán byl tedy XGBoost klasifikátor, jako nejúspěšnější

z množiny testovaných klasifikátorů. Následuje tabulka 3.1 optimálních hodnot hyperparametrů pro tento klasifikátor [39]. Názvy jsou uvedeny ve tvaru, ve kterém se zadávájí při konstrukci objektu klasifikátoru v programu.

Tabulka 3.1: Optimální hodnoty hyperparametrů klasifikátoru XGBoost. Subsample zde značí dílčí poměr trénovací instance (angl. Subsample Ratio of the Training Instance).

Název parametru	**Hodnota**	**Název parametru**	**Hodnota**
max_depth	7	learning_rate	0,1
n_estimators	1068	gamma	0
min_child_weight	1	colsample_bytre	0,65
subsample	0,9	colsample_bynode	0,6
reg_alpha	1e-08	reg_lambda	1

3.4 Struktura experimentu

Pro samotné experimenty v této práci, čili pro trénování a vyhodnocení výsledků klasifikátoru, byly využity metody balíčku **scikit-learn**. Nejprve je načten trénovací, případně i testovací dataset, jako objekt `DataFrame` z balíčku **pandas**. Následně je inicializován objekt `RepeatedStratifiedKFold` s počtem složek nastaveným na 10 a počtem opakování opět na 10. Tento objekt zastává roli generátoru train-test rozdělení pro metodu krosvalidace a jeho funkce je následující.

Nejprve vezme trénovací dataset a pro nastavení 10 složek vytvoří 10 stejně velikých podmnožin datasetu. Na 9 je natrénován klasifikátor a 10. je ponechána stranou pro validaci klasifikátoru. „Stratified“ znamená, že jsou jednotlivé třídy v jednotlivých složkách zastoupeny pokud možno vyrovnaně a nakonec „Repeated“ ve jméně znamená, že celý tento proces je n-krát zopakován, v našem případě desetkrát. Takto vytvořený objekt `RepeatedStratifiedKFold` je předán jako parametr funkci `cross_validate`, která vyhodnotí úspěšnost klasifikátoru, na základě sady metrik. Vyhodnocené úspěšnosti jsou zaznamenávány a nakonec jsou vráceny ve slovníku `scores`.

Metriky, které jsou předané funkci `cross_validate` jako parametr jsou definovány v konfiguračním souboru celého projektu diplomové práce, a sice

v souboru `config.json`, umístěném v kořenové složce projektu. Jednotlivé metriky jsou definovány v balíčku **scikit-learn** a použité byly následující: F1, Precision, Recall, Accuracy, Balanced_accuracy, ROC_AUC, Average_precision a Brier_score_loss. S tímto nastavením poté volána funkce `cross_validate`, která provede trénování a validaci klasifikátoru.

3.4.1 Trénovací a testovací data

Trénování bylo provedeno na daných promluvách od různých řečníků a v různých jazycích, konkrétně v češtině, němčině, angličtině a slovenštině. Konkrétně se trénovací dataset skládá z 88 promluv a testovací dataset se sestává z 20 promluv. Trénovací dataset se sestává z 98228 kandidátů pro které jsou vypočtené příznaky a testovací dataset obsahuje 20340 kandidátů. Trénovací dataset dále osahuje zhruba 24 minut nahrávek, zatímco testovací dataset jich obsahuje zhruba 5 minut.

3.5 Výsledný výchozí algoritmus

Takto sestavený experiment byl poté spuštěn ve virtuálním výpočetním prostředí Metacentrum. Výsledný výchozí algoritmus tedy zahrnuje skript pro předzpracování a natrénovaní klasifikátoru.

Po experimentu s trénováním klasifikátoru bylo provedeno testování na testovacích datech. Zde již nebylo využito k-složkové krosvalidace, byl pouze natrénován klasifikátor na všech trénovacích datech a následně byl ověřen na datech testovacích. Výsledky jsou shrnuty v tabulce 3.2. Pro trénování jsou uvedené hodnoty průměrem ze všech výsledků se směrodatnou odchylkou.

Tabulka 3.2: Úspěšnost výchozího algoritmu.

Metrika	**Výsledek validace** [%]	**Výsledek testu** [%]
F1	98,493 ± 0,126	98,209
Precision	98,509 ± 0,172	98,984
Avg. precision	99,902 ± 0,024	99,892
Recall	98,478 ± 0,191	97,446
Accuracy	98,281 ± 0,143	98,111
Bal. accuracy	98,249 ± 0,146	98,156
ROC AUC	99,876 ± 0.019	99,873
Brier score	0,013 ± 0.001	0,015

4 Rozšíření výchozího algoritmu

V této kapitole budou postupně představeny a vysvětleny hlavní přínosy, výsledky a postupy k nim vedoucí. Jako první krok k vylepšení algoritmu byl zvolen výběr a implementace nových příznaků, které by mohly přispět ke zlepšení úspěšnosti klasifikátoru. Ty jsou postupně vypsány v následující podkapitole. Hlavní myšlenkou bylo přidat co nejvíce nových příznaků a následně využít metod výběru příznaků pro určení jejich nejlepší podmnožiny.

4.1 Nové příznaky

Frekvence max. amplitudy v odhadu spektrální výkonové hustoty

Základ výpočtu tohoto příznaku tvoří Welchova metoda odhadu spektrální výkonové hustoty [60]. Tato metoda odhadu je implementována v balíčku **scipy** a nese jméno `welch`. Vstupní parametry jsou vzorky signálu a jeho vzorkovací frekvence. U tohoto příznaku bylo vybráno okolí 600 vzorků okolo každého kandidáta na hlasivkový puls. Toto okolí bylo zvoleno z důvodu pokrytí kontextu víceméně třech hlasivkových pulsů. V tomto okolí byl vypočítán odhad výkonové hustoty a následně byla určena maximální amplituda této funkce a k ní korespondující frekvence. Tato frekvence byla poté přidána do sloupečku datasetu s označením **WelchMaxFreq**.

Maximální amplituda v odhadu spektrální výkonové hustoty

Obdobným způsobem, jako v předešlém odstavci, byl vypočten i tento příznak. Namísto frekvence, odpovídající maximální amplitudě v odhadu spektrální výkonové hustoty, byla použita právě daná maximální amplituda. Ta byla poté zaznamenána do sloupečku datasetu **WelchMaxAmp**.

První formantová frekvence

Pro všechny následující příznaky, kde jsou vypočítávány formantové frekvence, bylo využito metody `formants` z balíčku **pyAudioAnalysis** [16]. Tato metoda opět přijímá jako parametr vzorky signálu a vzorkovací frekvenci. U každého kandidáta na hlasivkový puls bylo určeno okolí 600 vzorků a následně byly pomocí zmíněné metody vypočteny formantové frekvence.

Toto okolí je centrované kolem zkoumaného kandidáta. Z těch byla vybrána první a ta zaznamenána do datasetu do sloupečku **firstFormant**.

Druhá formantová frekvence

Stejným způsobem, jako v odstavci výše byla určena i druhá formantová frekvence a zaznamenána do sloupečku **secondFormant**.

Poměr první a druhé formantové frekvence

Po vypočtení první a druhé formantové frekvence bylo těchto znalostí využito pro výpočet jejich poměru. Ten by následně zaznamenán do sloupce **formantRatio**.

Vzdálenost první a druhé formantové frekvence

Obdobně byla vypočítána také vzdálenost první a druhé formantové frekvence, následně zaznamenána do sloupečku **formantDistance**.

Spectral Flux

Příznak spectral flux je opět počítán pro okolí 600 vzorků. Pro každého kandidáta na hlasivkový puls je vybráno dané okolí, které je předáno do funkce `stSpectralFlux` balíčku **pyAudioAnalysis**, spolu s okolím předešlého kandidáta. Pokud je zkoumán první kandidát je, jako předešlý vložen nulový signál. Příznak Spectral flux vyjadřuje spektrální změnu mezi dvěma následujícími signály, resp. dvěma okolími kandidátů na hlasivkový puls.

Peak slope

Tento příznak je počítán pomocí programu Octave, což je open source varianta programu a programovacího jazyka Matlab. Zdrojové kódy k použitým metodám jsou dostupné z [11]. K výpočtu tohoto příznaku byla použita metoda `peakslope`. Metoda výpočtu tohoto parametru byla představena v [23]. Pro programové volání funcí Ocatave přímo z interpretru Pythonu byl využit balíček **oct2py**, který poskytuje most mezi těmito dvěma programy. Následně tato metoda počítá peak slope parametr pro okolí kandidáta na hlasivkový puls, zvolené 1200 vzorků, to pokrývá víceméně šest kandidátů na hlasivkový puls a bylo nutné zvolit takovouto velikost, kvůli dostatečnému počtu vzorků, pro správný výpočet ve skriptu. Peak slope parametr je vrácen pro každého z šesti kandidátů. Následně je pak vypočítán průměr

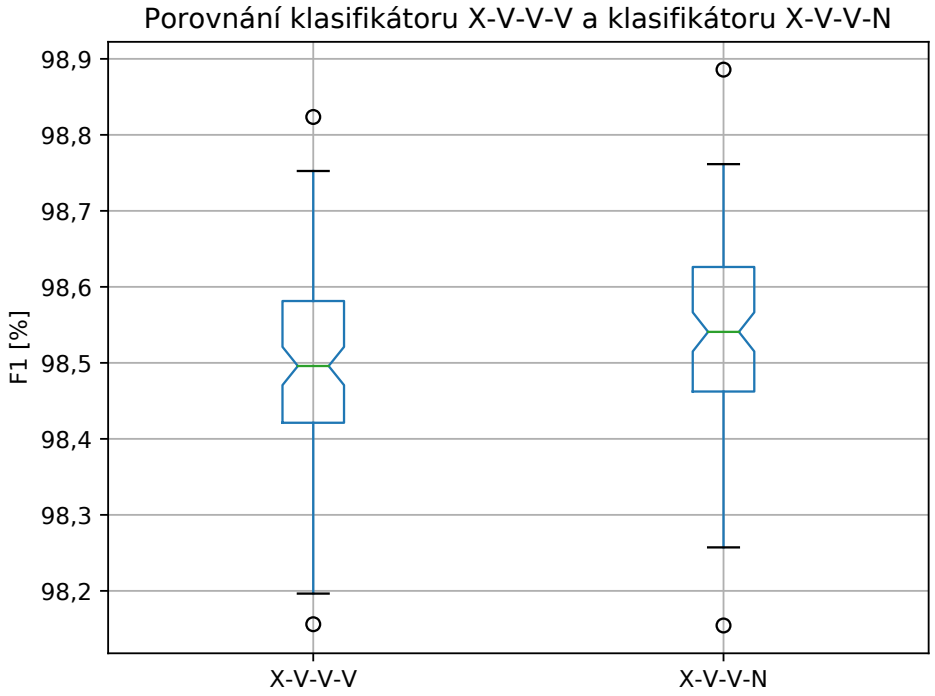
přes všechny vrácené hodnoty, který je pak vložen do datasetu do sloupce **PeakSlope**.

4.1.1 Vyhodnocení vlivu nových příznaků

Po přidání všech nových příznaků do datasetu byl proveden trénovací experiment ve stejném scénáři, jako v podkapitole 3.4. Kde je zvolena desetinásobná krosvalidace a desetisložkové vyvážené rozdělení datasetu do jednotlivých složek, při vyhodnocování dané množiny metrik. V následující tabulce 4.1 je porovnán výchozí algoritmus (**X-V-V-V**) a algoritmus s rozšířenou množinou příznaků (**X-V-V-N**). Zvýrazněny jsou ty výsledky, kde došlo ke zlepšení oproti výchozímu algoritmu. Na obrázku 4.1 je pak znázorněn Box and Whiskers graf znázorňující výsledky pro míru F1.

Tabulka 4.1: Porovnání výchozí klasifikátoru (X-V-V-V) a klasifikátoru s rozšířenou množinou příznaků (X-V-V-N).

Metrika	**X-V-V-V** [%]	**X-V-V-N** [%]
F1	98,493 ± 0,126	**98,541 ± 0,125**
Precision	98,509 ± 0,172	**98,564 ± 0,171**
Avg. prec.	99,902 ± 0,023	**99,907 ± 0,021**
Recall	98,478 ± 0,191	**98,516 ± 0,195**
Accuracy	98,281 ± 0,143	**98,335 ± 0,195**
Bal. accu.	98,249 ± 0,146	**98,305 ± 0,144**
ROC AUC	99,876 ± 0,019	**99,883 ± 0,018**
Brier score	0,013 ± 0,001	**0,013 ± 0,001**



Obrázek 4.1: Box and Whiskers graf výchozího klasifikátoru (X-V-V-V) a klasifikátoru s rozšířenou množinou příznaků (X-V-V-N).

4.1.2 Diskuze výsledků

V této kapitole byla původní množina příznaků rozšířena o nové příznaky, které, dle očekávání, přispěly ke zvýšení úspěšnosti klasifikace algoritmu, viz tabulku 4.1. Zde je patrný nárůst úspěšnosti ve všech metrikách. Podle obrázku 4.1 však ještě nemůžeme tvrdit, že došlo k statisticky významnému zlepšení, jelikož vyříznutí (angl. Notch) jednotlivých boxů se stále drobně překrývají. Dále tedy bude pokračováno v experimentech na rozšířené množině příznaků.

4.2 Další způsoby předzpracování

Dále bylo cílem nalezení nových metod předzpracování signálu, které by měly pozitivní vliv na úspěšnost klasifikátoru. Myšlen je zde proces před samotným vypočtením všech výše zmíněných příznaků. Původně je, jako metoda předzpracování, využita filtrace dolní propustí. Byly ověřeny dvě další metody, první z nich využívá tzv. MS signálu, který je počítán v al-

goritmu SEDREAMS a druhá využívá odstranění šumu metodou vlnkového prahování, obdobně, jako v [7]. Programově je zde využita knihovna `pywt` [34].

4.2.1 Mean-based signál

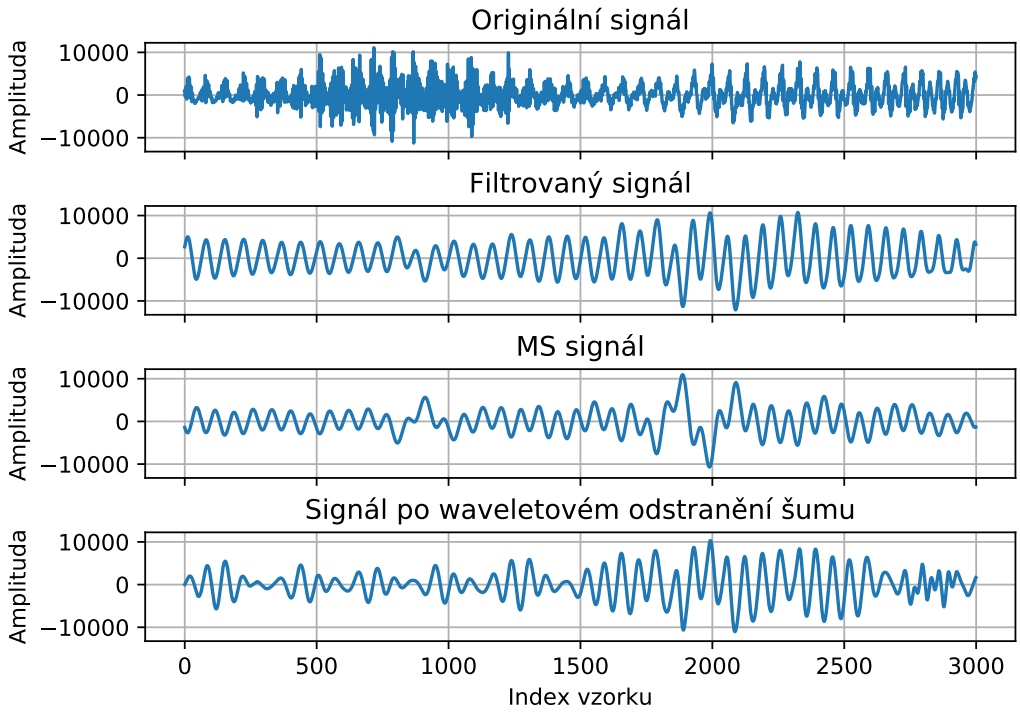
Pro transformaci vstupního signálu, na tzv. MS signál, je využita metoda `gci_sedreams` z [11] volaná z implementované metody `get_MS_sedreams`. V ní je postupováno podle návodu v [11]. Nejprve je určen pro daný zvukový soubor odhad základní hlasivkové frekvence a celý signál je následně přepolarizován kladně. Poté je přepolarizovaný signál, jeho vzorkovací frekvence, odhad základní hlasivkové frekvence a indikátor polarity vložen do funkce `gci_sedreams.m`, která vrací vypočtený Mean-based signál. Na něm jsou pak dále počítány všechny příznaky.

4.2.2 Waveletové prahování

Pro odstranění šumu metodou vlnkového prahování, nebo prahování pomocí vlnkové trasnformace, byla implementována metoda `denoise_signal`. V ní je nejprve určen práh (angl. Threshold), ze vstupního signálu. V základu je určen jako 63 % z největší amplitudy ve vstupním signálu se vyskytující. Následně jsou určeny koeficienty dekompozice signálu Waveletovou transformací pomocí funkce `wavedec`, která je obsažená v balíčku `pywt`. Každý koeficient je poté iterován a prahován s daným prahem funkcí `threshold`, s tím, že první koeficient je zahozen. Na konci je z prahovaných koeficientů rekonstruován signál metodou `waverec`.

4.2.3 Vyhodnocení

Na následujícím obrázku 4.2 jsou porovnány jednotlivé metody předzpracování v časové oblasti na výřezu vybrané nahrávky.

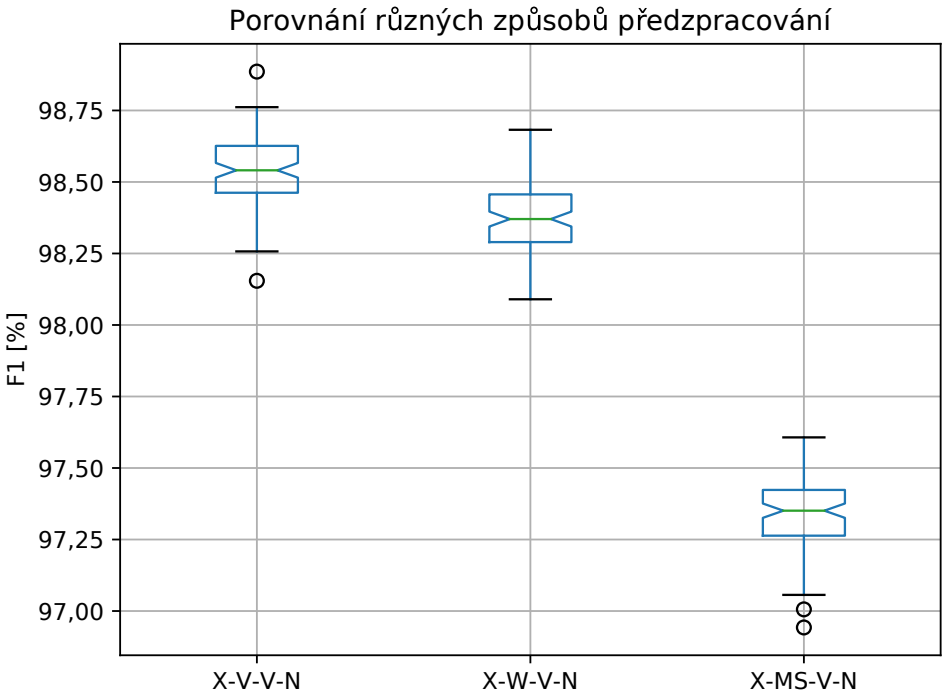


Obrázek 4.2: Porovnání různých metod předzpracování na stejném výřezu signálu.

Po vygenerování datasetů pomocí obou výše zmíněných metod předzpracování byly spuštěny experimenty podle již představeného scénáře v podkapitole 3.4, čili desetisložkové rozdělení a desetinásobná krosvalidace se všemi příznaky, tedy i nově přidanými. Výsledky jsou shrnuté v tabulce 4.2, pro klasifikátor s Mean-based předzpracováním (**X-MS-V-N**) a klasifikátor s předzpracováním pomocí vlnkové transformace (**X-W-V-N**). Pro porovnání je zde zahrnut i experiment s výchozím předzpracováním (**X-V-V-N**). Opět jsou zvýrazněné výsledky úspěšnějšího klasifikátoru. Na obrázku 4.3 je pak patrné statistické porovnání metriky F1 všech tří typů předzpracování.

Tabulka 4.2: Porovnání klasifikátoru s rozšířenou množinou příznaků (X-V-V-N), klasifikátoru s Mean-based předzpracováním (X-MS-V-N) a klasifikátoru s předzpracováním pomocí waveletové transformace (X-W-V-N).

Metrika	**X-V-V-N** [%]	**X-MS-V-N** [%]	**X-W-V-N** [%]
F1	**98,541 ± 0,125**	97,337 ± 0,141	98,375 ± 0,122
Precision	**98,564 ± 0,171**	96,145 ± 0,239	98,412 ± 0,154
Avg. prec.	**99,907 ± 0,021**	99,477 ± 0,052	99,895 ± 0,015
Recall	98,516 ± 0,195	**98,560 ± 0,158**	98,340± 0,172
Accuracy	98,335 ± 0,195	**98,441 ± 0,189**	97,578 ± 0,182
Bal. accu.	**98,305 ± 0,144**	95,443 ± 0,253	96,841 ± 0,253
ROC AUC	**99,883 ± 0,018**	99,087 ± 0,082	99,693 ± 0,038
Brier score	**0,013 ± 0,001**	0,031 ± 0,00162	0,01914 ± 0,00132



Obrázek 4.3: Box and Whiskers graf srovnávající výchozí algoritmus předzpracování (X-V-V-N), algoritmus předzpracován pomocí vlnkové transformace (X-W-V-N) a algoritmus předzpracování pomocí výstupního MS signálu z metody SEDREAMS (X-MS-V-N).

4.2.4 Diskuze výsledků

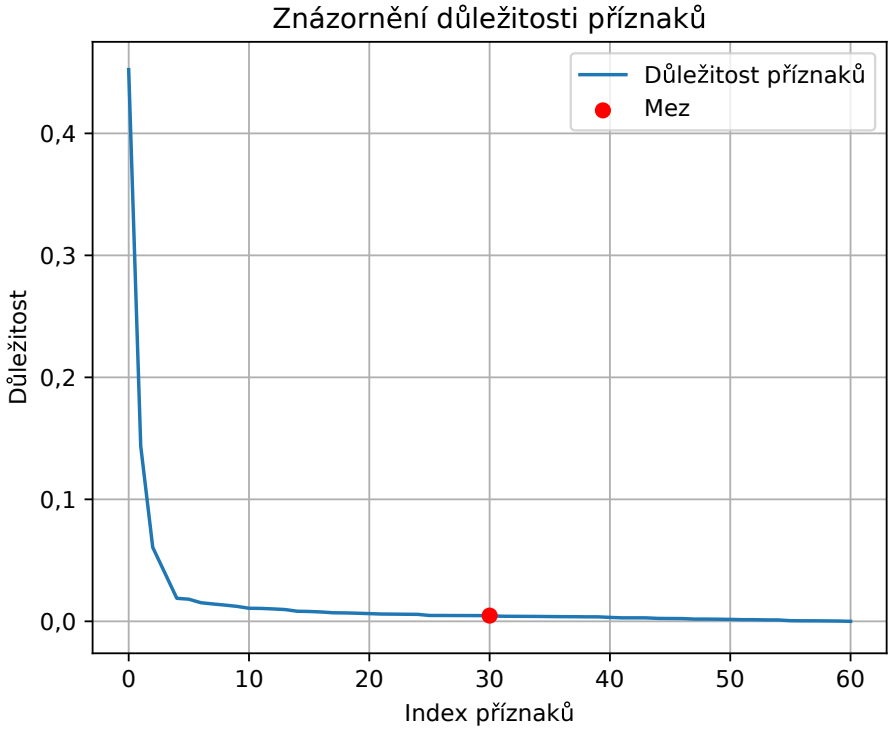
V této podkapitole byly na rozšířené množině příznaků z předchozího kroku experimentálně ověřeny dva jiné postupy předzpracování dat, v očekávání zlepšení úspěšnosti klasifikace. Podle výsledků experimentů, viz tabulku 4.2 a obrázek 4.3 je patrné, že ke kýženému zlepšení klasifikace nedošlo a výchozí způsob předzpracování nadále dosahuje statisticky významně lepších výsledků.

4.3 Výběr nejvhodnější podmnožiny příznaků

Po provedení experimentů s různým typem předzpracování a po přidání co největšího množství nových příznaků, přichází na řadu experimentální určení nejlepší podmnožiny ze všech příznaků.

4.3.1 Důležitost příznaků (angl. Feature Importance)

Jednou z výhod využití klasifikátoru založeného na souboru rozhodovacích stromů je, že je schopen sám podat odhad informativnosti jednotlivých příznaků. Pro tento účel je ve třídě `XGBClassifier` implementována členská proměnná `features_importances_`. Ta vrací slovník jednotlivých příznaků a jejich významnost. V tomto duchu byl sestaven první experiment pro výběr příznaků. Nejprve byl klasifikátor natrénován na všech příznacích a poté bylo prvních 30 vypsáno a seřazeno podle významnosti do následující tabulky 4.3. Toto číslo bylo určeno s ohledem na skutečnost, že po třicátém příznaku začne důležitost rychleji klesat k 0, viz obrázek 4.4. Zvýrazněné jsou ty příznaky, které byly v této práci nově přidány k výchozí množině. Míra významnosti, která je vrácená z natrénovaného klasifikátoru pro jednotlivé příznaky nemá přímou interpretaci, jedná se o vnitřní koeficienty klasifikátoru.



Obrázek 4.4: Znázornění důležitosti příznaků vzhledem k vybrané hranici pro zobrazení.

Tabulka 4.3: Tabulka významnosti příznaků podle XGBoost klasifikátoru.

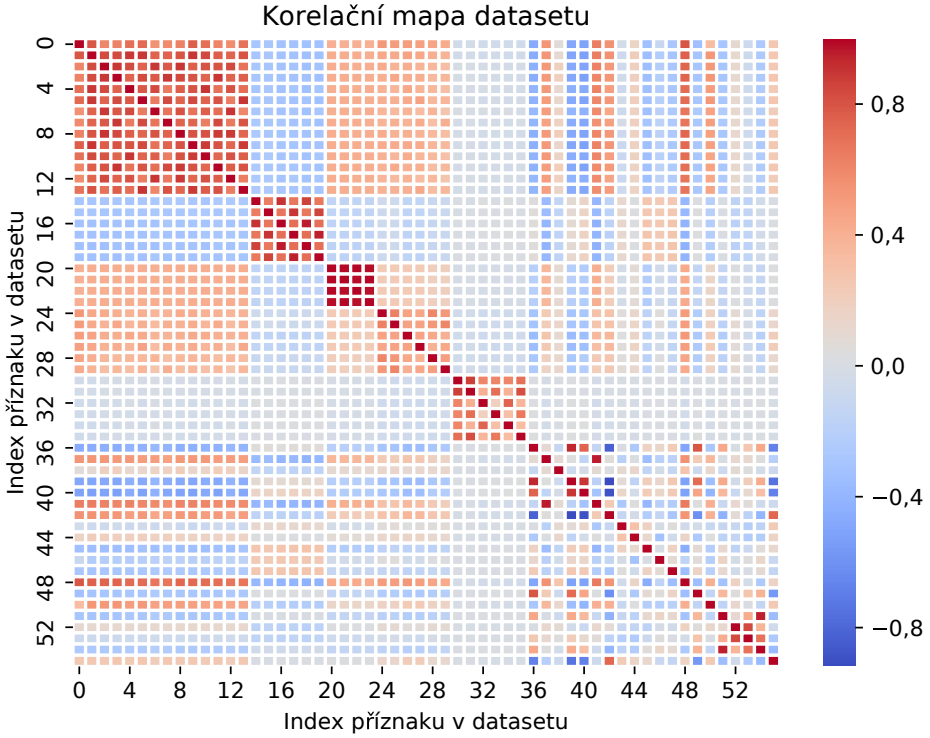
Název příznaku	**Významnost**	**Název příznaku**	**Významnost**
NegAmp0	0,4522	Zcr	0,0083
PosAmp-1	0,1437	TimeDif+3	0,0077
NegPeakRatio+1	0,0606	Mfcc1	0,007
NegPeakRatio-1	0,0398	**WelchMaxAmp**	0,0069
PosAmp-3	0,0189	Corr+2	0,0066
Corr-1	0,0181	Width+3	0,0063
NegAmp-2	0,0153	Hnr	0,006
Corr+1	0,0142	TimeDif+2	0,0059
F0	0,0133	PosAmp+3	0,0058
TimeDif-1	0,0123	NegAmp+3	0,0057
Corr-2	0,0107	**FirstFormant**	0,0048
TimeDif+1	0,0106	**WelchMaxFreq**	0,0048
NegAmp+2	0,0101	NegPeakRatio+2	0,0047
TimeDif-3	0,0097	Corr+3	0,0047
PosAmp+2	0,0083	**SecondFormant**	0,0047

4.3.2 Rekurzivní eliminace příznaků

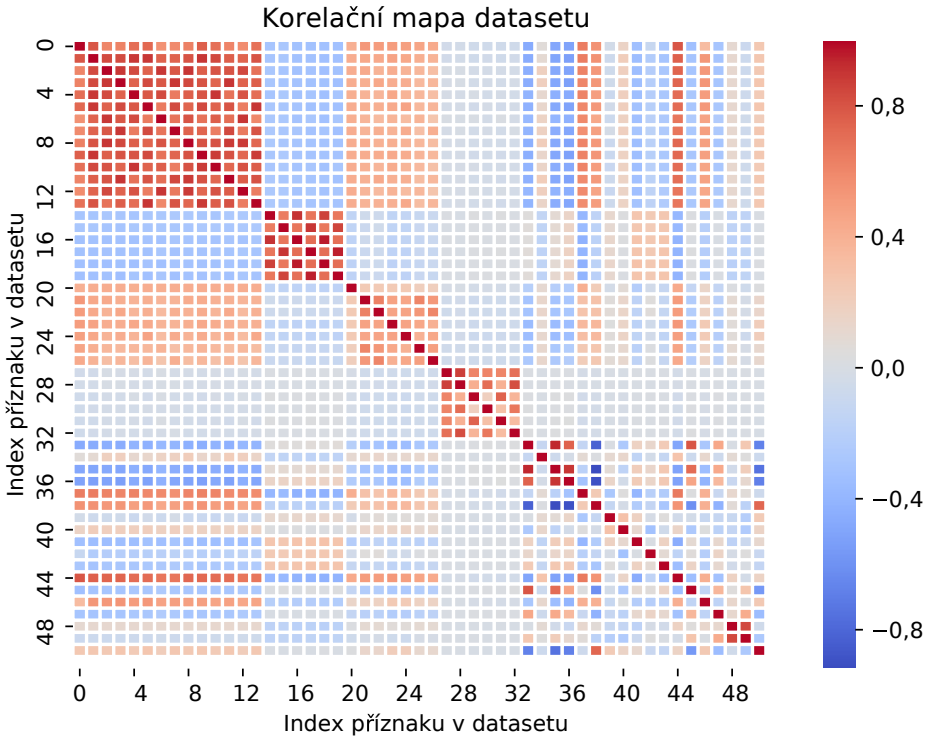
Následně bylo využito metody rekurzivní eliminace příznaků (angl. Recursive Feature Elimination, RFE), která transformuje výchozí množinu příznaků na její podmnožinu, podle specifikované míry. Tato metoda tedy vybere nejlepší počet příznaků a je schopna je i určit. Algoritmus funguje tak, že postupně vynechává jednotlivé příznaky a krosvalidací zjišťuje přínosnost tohoto vynechání. Iterativně pak tento postup opakuje, dokud nevyzkouší eliminaci všech příznaků. Implementaci této metody poskytuje balíček **scikit-learn**, konkrétně funkce `RFECV`. Experiment byl sestaven pro pětinásobnou krosvalidaci a jako metrika pro výběr příznaků byla zvolena metrika F1. Výstupní dataset byl posléze uložen do souboru.

4.3.3 Dekorelace transformované množiny příznaků

Výsledná transformovaná množina příznaků byla poté podrobena analýze korelovanosti jednotlivých příznaků. Byly vyloučeny ty příznaky, které vykazovaly velkou míru korelace a zároveň stály dostatečně nízko v žebříčku důležitosti, spočtené v odstavci 4.3.1. Pro vyjádření a vypočtení korelace jednotlivých příznaků byla použita knihovna **seaborn** [59], konkrétně metoda `heatmap`. Na následujících obrázcích je znázorněna transformovaná množina příznaků před dekorelací obrázek 4.5, a po ní, obrázek 4.6. Pro vyřazení nejvíce korelovaných příznaků, což znamená těch s mírou korelace vyšší než 0,9, bylo postupováno heuristicky. Nejprve tedy byly určeny příznaky, které odpovídají předešlé podmínce míry korelovanosti a následně byly iterativně odstraňovány při zohlednění poklesu úspěšnosti klasifikace, čili na základě jejich významnosti, viz tabulku 4.3. Proto nebyly odstraněny všechny tyto příznaky. Původně dataset se všemi příznaky obsahoval 66 příznaků, po selekci se tento počet snížil na 55 a po následné dekorelaci se podařilo toto číslo dále snížit na 50 příznaků. Odebrány byly následující příznaky: **secondFormant**, **energy**, **width+1**, **width-1** a **width+2**.



Obrázek 4.5: Korelační mapa příznaků po selekci.

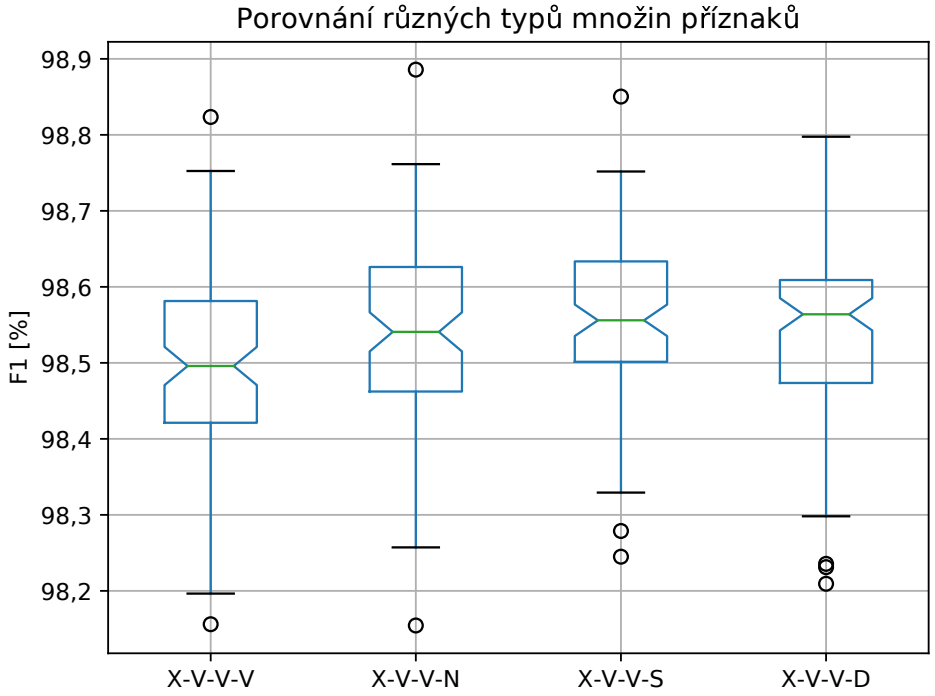


Obrázek 4.6: Korelační mapa příznaků po selekci a dekorelaci.

V následující tabulce 4.4 je možné shlédnout výsledky úspěšnosti klasifikátoru při trénování na množině vybraných příznaků (**X-V-V-S**) a na množině vybraných a dekorelovaných příznaků (**X-V-V-D**). Pro názornost jsou oba tyto případy porovnány s klasifikátorem natrénovaným na rozšířené množině příznaků (**X-V-V-N**). Na následujícím obrázku 4.7 je pak znázorněn Box and Whiskers graf výše zmíněných klasifikátorů a pro porovnání je přidán klasifikátor výchozí (**X-V-V-V**).

Tabulka 4.4: Porovnání klasifikátoru s rozšířenou množinou příznaků (X-V-V-N), klasifikátoru s vybranou podmnožinou příznaků (X-V-V-S) a klasifikátoru s vybranou a dekorelovanou podmnožinou příznaků (X-V-V-D).

Metrika	**X-V-V-N** [%]	**X-V-V-S** [%]	**X-V-V-D** [%]
F1	98,541 ± 0,125	**98,551 ± 0,11**	98,54 ± 0115
Precision	98,564 ± 0,171	98,533 ± 0,154	**98,566 ± 0,154**
Avg. prec.	**99,907 ± 0,021**	99,906 ± 0,021	99,906 ± 0,020
Recall	98,516 ± 0,195	98,535 ± 0,186	**98.543 ± 0,183**
Accuracy	98,335 ± 0,142	**98,347± 0,125**	98,335 ± 0,131
Bal. accu.	98,305 ± 0,144	**98,316 ± 0,126**	98,301 ± 0,134
ROC AUC	**99,883 ± 0,018**	99,882 ± 0,017	99,881 ± 0,018
Brier score	0,013 ± 0,001	**0,013 ± 0,001**	0,013 ± 0,00102



Obrázek 4.7: Box and Whiskers graf srovnávající výchozí klasifikátor (X-V-V-V), klasifikátor s rozšířenou množinou příznaků (X-V-V-N), klasifikátor s vybranou podmnožinou příznaků (X-V-V-S) a klasifikátor s vybranou a dekorelovanou podmnožinou příznaků (X-V-V-D).

4.3.4 Diskuze výsledků

Po neúspěchu s jinými druhy předzpracování se pozornost přesunula na zkvalitnění množiny příznaků. Nejprve byly vypsány jednotlivé příznaky s jejich významností z natrénovaného klasifikátoru XGBoost, viz tabulku 4.3. Zde je patrné, že některé nově vybrané příznaky se umístily v prvních třiceti z celkového počtu 66 příznaků. Po vybrání nejlepší podmnožiny příznaků, pomocí RFECV algoritmu, byla vykreslena tzv. korelační mapa, znázorňující korelaci mezi jednotlivými příznaky, viz obrázek 4.5. V tabulce 4.4 je patrné zvýšení úspěšnosti klasifikátoru nejprve pro vybranou podmnožinu a následně víceméně udržení této úspěšnosti i pro dekorelovanou a vybranou podmnožinu. Na obrázku 4.7 je pak patrné, díky nepřekrytí výřezů boxů, že klasifikátor natrénovaný na takto vybrané podmnožině a především vybrané a dekorelované podmnožině je již statisticky významně úspěšnější, než výchozí algoritmus, a proto byla dále použita vybraná a dekorelovaná

podmnožina příznaků.

4.4 Výběr nových hodnot hyperparametrů na základě nejvhodnější podmnožiny příznaků

Po nalezení suboptimální podmnožiny příznaků, bylo, jako další experiment, provedeno nalezení nových hyperparametrů klasifikátoru XGBoost. V této podkapitole bylo čerpáno z knihy [4]. Opět bylo využito balíčku **sklearn**, tentokrát metody `GridSearchCV` s trojnásobnou krosvalidací. Tato metoda iterativně prochází všechny kombinace hyperparametrů, definované ve slovníku, který je do ní vložen, a pro každou kombinaci provádí trénování a trojnásobnou krosvalidaci. Postupně zaznamenává vypočtené úspěšnosti a pro nejvyšší z nich pak vrací její hodnotu a natrénovaný objekt klasifikátoru. Použité hodnoty jsou uvedeny v tabulce 4.5. Vybrané hodnoty byly určeny heuristicky a s pomocí informací z [4]. V tabulce nejsou uvedeny výchozí hodnoty hyperparametrů, zde bylo cíleno na nalezení nových lepších hodnot hyperparametrů. Tučně vyznačené jsou ty parametry, které algoritmus vybral.

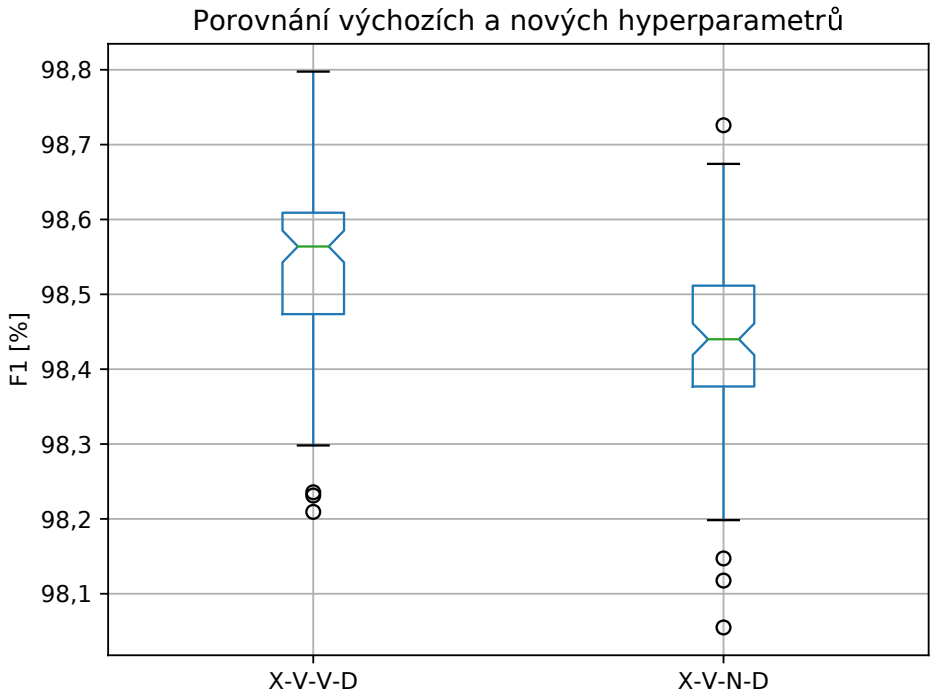
Tabulka 4.5: Použité hodnoty pro hledání nových hyperparametrů klasifikátoru. „Subsample“ zde značí dílčí poměr trénovací instance (angl. Subsample Ratio of the Training Instance).

Název parametru	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Max_depth	6	7	**8**	9			
Learning_rate	**0,01**	0,05	0,1	0,125			
N_estimators	1000	1050	1100	1150	1200	1250	**1300**
Subsample	0,5	0,6	0,8	**0,9**			

V následující tabulce 4.6 je srovnání úspěšnosti klasifikátoru s výchozími hodnotami hyperprametrů z [39], vybranou a dekorelovanou množinou příznaků (**X-V-V-D**) a klasifikátoru s novými hyperparametry a stejnou množinou příznaků (**X-V-N-D**).

Tabulka 4.6: Porovnání klasifikátoru s vybranou a dekorelovanou podmnožinou příznaků (X-V-V-D) s klasifikátorem se stejnou množinou příznaků, ale novými hyperparametry (X-V-N-D).

Metrika	**X-V-V-D** [%]	**X-V-N-D** [%]
F1	**98,541 ± 0,125**	98,433 ± 0,118
Precision	**98,564 ± 0,171**	98,410 ± 0,171
Avg. prec.	**99,907 ± 0,021**	99,897 ± 0,018
Recall	**98,516 ± 0,195**	98,445 ± 0,193
Accuracy	**98,335 ± 0,142**	98,212 ± 0,134
Bal. acu.	**98,305 ± 0,144**	98,172 ± 0,137
ROC AUC	**99,883 ± 0,018**	99,867 ± 0,018
Brier score	0,013 ± 0,010	**0,013 ± 0,001**



Obrázek 4.8: Box and Whiskers graf srovnání klasifikátoru s vybranou a dekorelovanou podmnožinou příznaků (X-V-V-D) s klasifikátorem se stejnou množinou příznaků, ale novými hyperparametry (X-V-N-D).

4.4.1 Diskuze výsledků

V této podkapitole následoval pokus s nalezením nových hodnot hyperparametrů klasifikátoru, proto prohledávací tabulka záměrně neobsahovala původní hodnoty hyperparametrů. Po prohledání definované tabulky hodnot, viz tabulku 4.5, byly nalezeny nové hodnoty hyperparametrů. Jak se ale ukázalo při ověření úspěšnosti klasifikátoru, oproti výchozím hodnotám hyperparametrů, viz tabulku 4.6 a obrázek 4.8, nedošlo ke zlepšení úspěšnosti, a proto byly pro další pokusy ponechány výchozí hyperparametry. Teoreticky by šlo dále zvětšovat počet estimátorů, čili parametr n_estimators, to by však postupně vedlo k přetrénování klasifikátoru, viz článek [4].

4.5 Porovnání vylepšeného algoritmu s předzpracováním a výběrem příznaků pomocí konvoluční neuronové sítě

Výsledný nejlepší algoritmus, to znamená XGBoost klasifikátor s výchozími hyperparametry a rozšířenou, vybranou a dekorelovanou množinou příznaků, byl v dalším experimentu porovnán s novým typem předzpracování. Jedná se o trénovací a testovací dataset, který je generován pomocí konvoluční neuronové sítě, jejíž architektura byla inspirována [54].

Kompletní výpis použité architektury konvoluční neuronové sítě, spolu s popisem trénování je v příloze B. Jako vstup byly využity nepředzpracované promluvy, vzorkované 8kHz vzorkovací frekvencí. Konvoluční neuronová síť má tu vlastnost, že je schopna sama provést výběr a selekci příznaků, vhodných pro zpracování v dalších vrstvách sítě [15]. Tyto příznaky jsou získány jako výstup z poslední husté (angl. Dense) vrstvy a poslední konvoluční vrstvy. Výsledné vektory příznaků mají pak délku 64 a 6656.

Pro porovnání byl proveden testovací experiment současného nejlepšího klasifikátoru XGBoost (**X-8-V-D**) a klasifikátorů XGBoost natrénovaných a testovaných na datech z konvoluční neuronové sítě pro 64 a 6656 příznakové vektory (**X-C1-V-D**) a (**X-C2-V-D**). Nejprve byly natrénovány klasifikátory na všech trénovacích datech a následně byly validovány na separátních testovacích datech. V následující tabulce 4.7 jsou porovnány úspěšnosti jednotlivých klasifikátorů. Zvýrazněn je vždy výsledek nejúspěšnějšího klasifikátoru. Nutno podotknout, že neuronové sítě byly trénovány a testovány na 8 kHz datech, bylo tedy nutné to samé provést pro XGBoost klasifikátor. Zde ovšem byla použita původní data vzorkovaná 16 kHz vzorkovací frekvencí, pouze byly brány v potaz 8 kHz referenční hodnoty o hlasivkových pulsech.

Tím bylo docíleno stejného rozměru predikovaných vektorů a následně bylo možné provést McNemarův test, pro statistické porovnání klasifikátorů.

Tabulka 4.7: Provnání úspěšnosti současného nejlepšího klasifikátoru XGBoost (X-8-V-D) a klasifikátorů XGBoost natrénovaných a testovaných na datech z konvoluční neuronové sítě pro 64 a 6656 příznakové vektory (X-C1-V-D a X-C2-V-D).

Metrika	**X-8-V-D** [%]	**X-C1-V-D** [%]	**X-C2-V-D** [%]
F1	98,294	98,283	**98,481**
Precision	**99,023**	98,576	98,871
Avg. prec.	99,903	99,864	**99,907**
Recall	97,576	97,992	**98,094**
Accuracy	97,875	97,852	**98,102**
Bal. acu.	97,977	97,804	**98,105**
ROC AUC	99,833	99,769	**99,841**
Brier score	0,017	0,019	**0,016**

Následně byly provedeny McNemarovy testy pro jednotlivé kombinace klasifikátorů. Prvním byl test pro nulovou hypotézu, že pravděpodobnost správné klasifikace X-8-V-D klasifikátoru je statisticky stejná, jako pravděpodobnost správné klasifikace X-C1-V-D klasifikátoru. Alternativní hypotéza poté je, že se liší. Následuje kontingenční tabulka 4.8, která byla pro McNemarovo test spočtena pomocí balíčku **mlxtend** [51].

Tabulka 4.8: Kontingenční tabulka McNemarova testu pro srovnání pravděpodobnosti správné klasifikace současného nejlepšího klasifikátoru XGBoost (X-8-V-D) a klasifikátorů XGBoost natrénovaných a testovaných na datech z konvoluční neuronové sítě pro 64 příznakové vektory (X-C1-V-D).

	Úspěch X-C1-V-D	**Selhání X-C1-V-D**
Úspěch X-8-V-D	16688	173
Selhání X-8-V-D	169	197

$$\chi^2 = 0{,}026$$
$$p = 0{,}871$$

Následoval McNemarův test pro nulovou hypotézu, že pravděpodobnost správné klasifikace pro klasifikátor X-C2-V-D je stejná, jako pro klasifikátor X-C1-V-D. Alternativní hypotéza byla, že se pravděpodobnost správné klasifikace pro oba klasifikátory liší.

Tabulka 4.9: Kontingenční tabulka pro McNemarův test pro klasifikátory XGBoost natrénované a testované na datech z konvoluční neuronové sítě pro 64 a 6656 příznakové vektory (X-C1-V-D a X-C2-V-D).

	Úspěch X-C2-V-D	**Selhání X-C2-V-D**
Úspěch X-C1-V-D	16797	60
Selhání X-C1-V-D	103	267

$$\chi^2 = 10{,}822$$
$$p = 0{,}001$$

Nakonec byl proveden McNemarův test pro nulovou hypotézu, že pravděpodobnost správné klasifikace klasifikátoru X-8-V-D je stejná, jako pro klasifikátor X-C2-V-D. Alternativní hypotéza byla, že se pravděpodobnost správné klasifikace pro oba klasifikátory liší.

Tabulka 4.10: Kontingenční tabulka McNemarova testu pro srovnání současného nejlepšího klasifikátoru XGBoost (X-8-V-D) a klasifikátorů XGBoost natrénovaných a testovaných na datech z konvoluční neuronové sítě pro 6656 příznakové vektory (X-C2-V-D).

	Úspěch X-C2-V-D	**Selhání X-C2-V-D**
Úspěch X-8-V-D	16733	128
Selhání X-8-V-D	167	199

$$\chi^2 = 4{,}895$$
$$p = 0{,}027$$

4.5.1 Diskuze výsledků

V této podkapitole byl nejlepší vylepšený algoritmus (X-8-V-D) porovnán s jiným druhem předzpracování, a sice s předzpracováním pomocí konvoluční neuronové sítě. Toto předzpracování lze rozdělit do dvou typů, s velikostí vektoru příznaků 64 (X-C1-V-D) a 6656 prvků (X-C2-V-D). Po provedení experimentů byl zjištěn nárůst úspěšnosti klasifikace pro předzpracování typu CNN 6656 (X-C2-V-D), viz tabulku 4.7. Pro všechny tři kombinace klasifikátorů byly vyčísleny McNemarovy testy, viz tabulky 4.8, 4.9 a 4.10. V McNemarově testu, viz tabulku 4.8, vyšlo, že na hladině významnosti $\alpha = 0{,}05$ jsou pravděpodobnosti správné klasifikace obou klasifikátorů podobné, čili přijímáme nulovou hypotézu. V dalším testu 4.9 bylo vyčísleno, že na hladině významnosti $\alpha = 0{,}05$ jsou pravděpodobnosti správné klasifikace klasifikátorů rozdílné, čili zamítáme nulovou hypotézu a za statisticky významně úspěšnější z těchto dvou klasifikátorů lze označit klasifikátor s předzpracováním typu CNN 6656 (X-C2-V-D). To samé lze vyčíst i z posledního testu 4.10, kde jsou opět na hladině významnosti $\alpha = 0{,}05$ jsou pravděpodobnosti správné klasifikace rozdílné, čili zamítáme nulovou hypotézu. Za statisticky úspěšnější lze tedy označit klasifikátor s předzpracováním typu CNN 6656 (X-C2-V-D), a proto bude zahrnut do následujících experimentů spolu s X-V-V-D klasifikátorem.

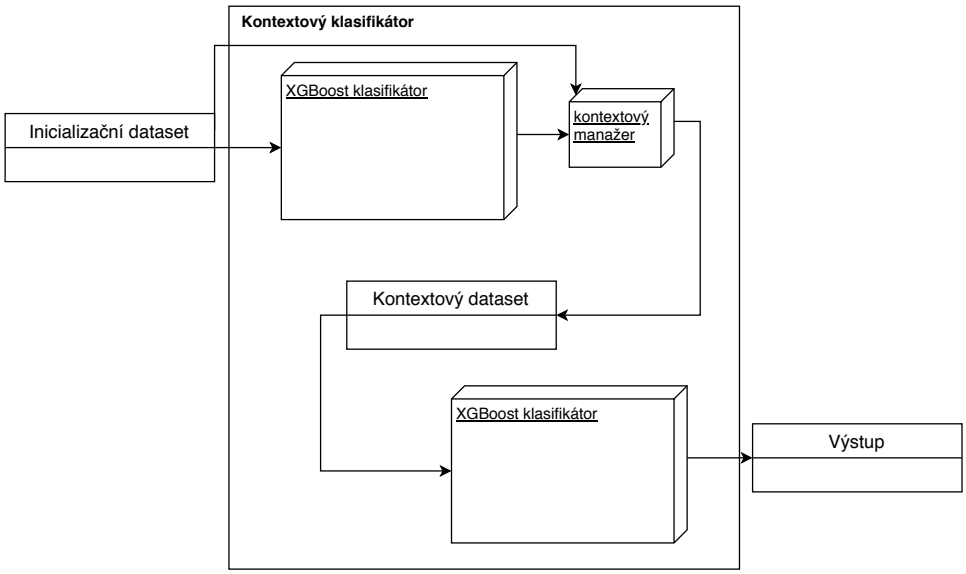
4.6 Kontextový klasifikátor

V této kapitole bude představen koncept a následně, implementace a ověření tzv. kontextového klasifikátoru (angl. Context Aware Classifier). Ten je blíže popsán také v článku [40]. Hlavní motivací tohoto algoritmu je další zlepšení úspěšnosti detekce hlasivkového pulsu, díky zavedení informace o širším kontextu okolních kandidátů.

4.6.1 Struktura klasifikátoru

Hlavní stavební jednotku tvoří opět XGBoost klasifikátor. Zde jsou využity dva tyto klasifikátory, spojené v sérii za sebou, kdy z predikce prvního je sestaven tzv. kontextový dataset a ten předán jako vstup do druhého klasifikátoru. Hodnoty hyperparametrů jsou použité stejné jako doposud a u obou klasifikátorů stejné. Jsou implementovány dvě možnosti vytvoření kontextového datasetu. První možností je kontextový dataset sestavit pouze z predikcí okolních pravděpodobností hlasivkových pulsů s volitelným kontextem n kandidátů, centrovaným okolo aktuálního kandidáta. Druhou možností je k výše zmíněnému přidání libovolných již známých příznaků z výchozího datasetu. Architektura je ilustrována na obrázku 4.9 níže. Výsledkem je pak již binární informace o jednotlivých kandidátech, zda se jedná o hlasivkový puls, či nikoliv.

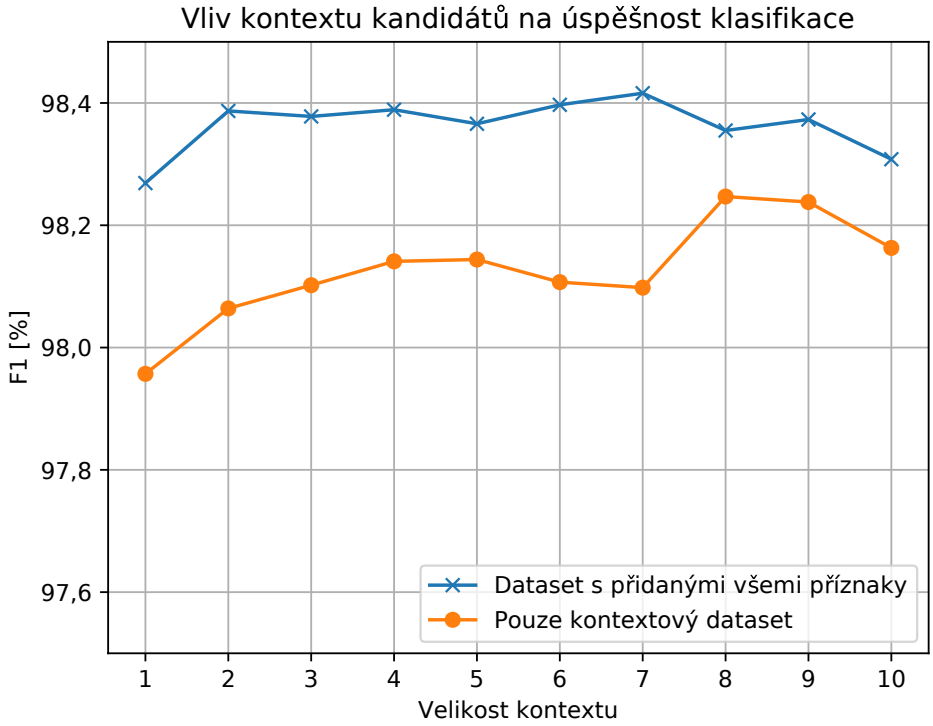
Vnitřní chování klasifikátoru je následující a využívá principu tzv. mimosložkové trénování (angl. Out of Fold). Po vstupu inicializačního datasetu je rozdělen na deset složek, jeden je ponechán nedotčený pro testování. Na devíti složkách je inicializační klasifikátor natrénován a následně je testován na ponechané desáté složce. Výsledek po testu je zaznamenán do kontextového datasetu. Iterováno je tak, aby testovací složka postupně prošla všechny řádky inicializačního datasetu a aby byl vyplněn kontextový dataset stejné délky. Následně je v kontextovém manažeru vytvořen kontext, čili jsou do dané řádky přidány hodnoty daného počtu předchozích a následujících kandidátů na hlasivkový puls. Volitelně jsou v kontextovém manažeru vloženy dané indexy příznaků z inicializačního datasetu. Na takto zpracovaném kontextovém datasetu je natrénován kontextový klasifikátor.



Obrázek 4.9: Architektura kontextového klasifikátoru.

4.6.2 Experimenty

V první sadě experimentů byl klasifikátor nastaven tak, aby do kontextového datasetu přidával pouze odhad pravděpodobnosti aktuálního kandidáta na hlasivkový puls. V kontextovém manažeru se pak sloučí do jedné řádky daný počet předchozích a budoucích kandidátů a vytvoří se tak kontext. Takto bylo postupováno pro kontext 1-10. Klasifikátor byl vždy nacvičen na všech trénovacích datech a následně testován na separátních testovacích. Na následujícím obrázku 4.10 je patrná závislost délky kontextu na úspěchu klasifikace, pro míru F1. V druhé sadě experimentů byly ke kontextu přidány všechny příznaky pro daného kandidáta, které obsahoval inicializační dataset, tedy stejný dataset, jako například pro klasifikátor X-V-V-D. Opět byly provedeny experimenty pro kontext 1-10. A na stejném obrázku 4.10 je také patrná závislost úspěšnosti klasifikátoru na velikosti kontextu. Pro přehlednost jsou výsledky znázorněné v tomto obrázku vypsány do následující tabulky 4.11. Oba experimenty byly provedeny pro klasické příznaky, nikoliv pro předzpracování konvoluční neuronové sítě.



Obrázek 4.10: Závislost velikosti kontextu na úspěšnosti klasifikace, pro případ přidání všech příznaků a pro čistě kontextový dataset.

Tabulka 4.11: Porovnání úspěšnosti klasifikace v závislosti na velikosti kontextu, pro případ přidání všech příznaků (KAn-V-V-NVD) a pro čistě kontextový dataset (Kn-V-V-NVD). Použita byla míra F1.

Kontext n	**Kn-V-V-NVD** [%]	**KAn-V-V-NVD** [%]
1	97,957	98,269
2	98,064	98,387
3	98,102	98,378
4	98,141	98,389
5	98,144	98,366
6	98,107	98,397
7	98,098	98,416
8	98,247	98,355
9	98,238	98,373
10	98,163	98,308

Pro nejlepší klasifikátor z předchozích experimentů, viz tabulku 4.11, tedy pro kontextový klasifikátor s kontextem 7 a všemi přidanými příznaky (**KA7-V-V-D**). Byly vyhodnoceny všechny míry, viz tabulku 4.12, kde je pro porovnání uveden i doposud nejlepší klasický klasifikátor, čili XGBoost s klasickými příznaky (klasické příznaky značí použití výchozích metod předzpracování a extrakce, nikoliv nový způsob pomocí konvoluční neuronové sítě) a jejich rozšířenou, vybranou a dekorelovanou podmnožinou (**X-V-V-D**).

Tabulka 4.12: Porovnání úspěšnosti klasifikátoru XGBoost s klasickými příznaky (X-V-V-D) a kontextový klasifikátor s kontextem 7 a všemi přidanými příznaky (KA7-V-V-D).

Metrika	**X-V-V-D** [%]	**KA7-V-V-D** [%]
F1	98,261	**98,416**
Precision	99,032	**99,053**
Avg. prec.	**99,923**	99,913
Recall	97,502	**97,788**
Accuracy	98,166	**98,328**
Bal. accu.	98,210	**98,364**
ROC AUC	99,885	**99,898**
Brier score	0,014	0,014

Následně byl vyčíslen McNemarův test pro nejlepší kontextový klasifikátor (KA7-V-V-D) a pro klasický XGBoost klasifikátor (X-V-V-D). Nulová hypotéza tedy je, nechť pravděpodobnost správné klasifikace nejlepšího kontextového klasifikátoru je stejná jako pravděpodobnost správné klasifikace klasického XGBoost klasifikátoru. Alternativní hypotéza je, že se tyto pravděpodobnosti liší.

Tabulka 4.13: Kontingenční tabulka pro McNemarův test pro nejlepší kontextový klasifikátor (KA7-V-V-D) a dosavadní nejlepší XGBoost (X-V-V-D).

	Úspěch KA7-V-V-D	**Selhání KA7-V-V-D**
Úspěch X-V-V-D	19918	47
Selhání X-V-V-D	80	293

$$\chi^2 = 8{,}063$$
$$p = 0{,}005$$

Obdobně, jako na obrázku 4.10 a v tabulce 4.11, byla určena nejlepší velikost kontextu pro klasifikátor s předzpracováním typu CNN 6656. Z důvodu výpočetní náročnosti byly provedeny pouze dva experimenty v okolí maxima na obrázku 4.11 a pouze pro kontext se všemi příznaky, jak bylo ověřeno výše. Výsledky jsou uvedeny v následující tabulce 4.14.

Tabulka 4.14: Porovnání úspěšnosti klasifikace v závislosti na velikosti kontextu, pro případ přidání všech příznaků s předzpracováním typu CNN 6656. Použita byla míra F1.

Kontext	**Kontextový dataset s příznaky**
6	98,524
7	98,483

Pro kontext 6 okolních kandidátů se všemi příznaky (**KA6-C2-V-D**) byly vyčísleny všechny použité metriky a pro porovnání jsou sepsané spolu s původním klasifikátorem typu CNN 6656 (**X-C2-V-D**) v následující tabulce 4.15. Následně byl proveden McNemarův test, viz tabulku 4.16, pro porovnání kontextového CNN 6656 klasifikátoru a původního CNN 6656 klasifikátoru. Nulová hypotéza tedy je, že pravděpodobnost správné klasifikace nejlepšího kontextového klasifikátoru s předzpracováním typu CNN 6656 je stejná, jako pravděpodobnost správné klasifikace původního XGBoost klasifikátoru s předzpracováním typu CNN 6656. Alternativní hypotéza je, že se tyto pravděpodobnosti liší.

Tabulka 4.15: Porovnání úspěšnosti klasifikátoru s předzpracováním typu CNN 6656 (X-C2-V-D) a kontextového klasifikátoru s předzpracováním typu CNN 6656 (KA6-C2-V-D).

Metrika	**X-C2-V-D** [%]	**KA6-C2-V-D** [%]
F1	98,481	**98,524**
Precision	**98,871**	98,836
Avg. prec.	99,907	**99,909**
Recall	98,094	**98,214**
Accuracy	98,102	**98,154**
Bal. accu.	98,105	**98,134**
ROC AUC	99,841	**99,842**
Brier score	0,016	0,016

Tabulka 4.16: Kontingenční tabulka pro McNemarův test pro nejlepší kontextový CNN 6656 klasifikátor se všemi příznaky (KA6-C2-V-D) a dosavadní nejlepší CNN 6656 (X-C2-V-D).

	Úspěch X-C2-V-D	**Selhání X-C2-V-D**
Úspěch KA6-C2-V-D	16867	42
Selhání KA6-C2-V-D	33	285

$$\chi^2 = 0{,}853$$
$$p = 0{,}356$$

4.6.3 Porovnání kontextového klasifikátoru s ostatními algoritmy

V této podkapitole budou klasifikátory KA7-V-V-D a X-V-V-D porovnány vůči ostatním používaným algoritmům. Následující tabulka 4.17 je převzata z článku [40]. Algoritmy byly testované na čtyřech datasetech. Prvním byl UWB, ten značí využití dat stejných, jako používané v této práci. Dále byl použit dataset BDL, ten obsahuje data od mužského řečníka z USA. Následně byl použit dataset SLT, ten obsahuje data od ženského řečníka z USA, a jako poslední byl použit dataset s označením KED TIMIT, viz [1]. Detailnější popis použitých algoritmů je představen v článku [40]. Jednotlivé porovnávané algoritmy, včetně vyhodnocovacích metrik, byly představeny v úvodní podkapitole 2.7, resp. sekci 2.5.2.

Tabulka 4.17: Souhrn úspěšnosti klasifikačních algortmů pro detekci hlasivkových pulsů na čtyřech datasetech, převzato z [40], Užité metriky jsou zobrazeny v procentech, kromě metriky IDA, jejíž jednotkou jsou milisekundy.

Dataset	**Method**	**IDR**	**MR**	**FAR**	**IDA**	**A25**	**E10**
UWB	KA7-V-V-D	**96,78**	**2,34**	0,88	0,24	98,73	**95,66**
	X-V-V-D	96,60	2,56	**0,85**	0,23	98,79	95,54
	SEDREAMS	93,12	4,00	2,88	0,28	98,10	91,69
	MMF	85,08	11,43	3,48	0,47	97,85	83,55
	DYPSA	89,64	6,25	4,11	0,37	98,04	88,22
	REAPER	92,81	5,51	1,69	0,27	98,00	91,45
	GEFBA	91,24	7,68	1,08	**0,22**	**98,89**	90,34
	PSFM	88,17	9,71	2,12	0,39	98,27	86,88
BDL	KA7-V-V-D	**94,19**	**2,80**	3,01	0,37	98,59	**92,90**
	X-V-V-D	94,04	2,93	3,03	**0,36**	98,58	92,74
	SEDREAMS	91,80	3,03	5,16	0,45	97,37	90,02
	MMF	90,42	4,63	4,95	0,56	97,15	87,87
	DYPSA	89,43	4,38	6,19	0,54	97,13	86,89
	REAPER	93,24	4,39	2,37	0,56	98,01	91,47
	GEFBA	87,93	10,05	**2,02**	1,02	**99,11**	87,18
	PSFM	87,05	9,65	3,30	0,71	96,95	84,50
SLT	KA7-V-V-D	**96,64**	1,34	2,01	**0,17**	99,73	**96,39**
	X-V-V-D	96,49	1,57	**1,95**	0,19	99,71	96,22
	SEDREAMS	94,66	**1,13**	4,21	0,17	99,67	94,36
	MMF	92,44	5,29	2,26	0,40	99,17	91,78
	DYPSA	93,25	2,91	3,84	0,32	99,39	92,75
	REAPER	95,57	1,66	2,77	0,19	99,67	95,27
	GEFBA	94,85	2,62	2,53	**0,17**	**99,76**	94,63
	PSFM	86,95	10,46	2,60	0,45	99,26	86,42
KED	KA7-V-V-D	**96,82**	**2,31**	0,87	0,24	98,63	**95,83**
	X-V-V-D	96,60	2,56	0,85	0,22	98,76	95,68
	SEDREAMS	92,30	6,03	1,66	0,29	99,12	91,76
	MMF	90,16	7,16	2,68	0,35	98,99	89,52
	DYPSA	90,27	7,07	2,65	0,30	99,25	89,72
	REAPER	91,05	8,18	**0,78**	0,28	99,47	90,67
	GEFBA	88,51	10,36	1,13	**0,21**	**99,74**	88,30
	PSFM	89,47	9,59	0,94	0,39	99,22	88,85

4.6.4 Diskuze výsledků

V hlavní části této práce byl implementován algoritmus kontextového klasifikátoru. Ten byl následně ověřen pro kontexty délky 1-10 a pro dvě konfigurace. První ověřená konfigurace byla pro kontextový dataset, jenž byl složen jen z pravděpodobností kandidátů na hlasivkový puls, a druhá konfigurace k tomuto přidala všechny příznaky pro daný hlasivkový puls z inicializačního datasetu. Závislosti úspěšnosti kontextového klasifikátoru na velikosti kontextu jsou pro obě konfigurace znázorněny na obrázku 4.10. Podle tohoto obrázku lze určit, že nejlepší velikost kontextu pro všechny přidané příznaky se rovná 7. Kontextový klasifikátor s kontextem 7 (KA7-V-V-D) a se všemi přidanými příznaky byl porovnán s původním nejlepším klasickým klasifikátorem (X-V-V-D) v tabulce 4.12 a poté byl vyčíslen McNemarův test, viz tabulku 4.13. V něm bylo prokázáno, že na hladině významnosti $\alpha = 0{,}05$ je kontextový klasifikátor úspěšnější, než původní klasifikátor. Obdobně byl určen nejlepší kontext pro kontextový klasifikátor s předzpracováním typu CNN 6656, viz tabulku 4.14. Ten byl stanoven na 6 (KA6-C2-V-D). Poté byly pro tento klasifikátor určeny všechny metriky a byl porovnán s původním klasifikátorem s předzpracováním typu CNN 6656 (X-C2-V-D) v tabulce 4.15. Následně byl vyčíslen McNemarův test pro porovnání obou těchto klasifikátorů, viz tabulku 4.16. Z něj je patrné, že na hladině významnosti $\alpha = 0{,}05$ není kontextový klasifikátor KA6-C2-V-D úspěšnější, než jeho původní podoba. V poslední řadě byly klasifikátor KA7-V-V-D a X-V-V-D porovnány na čtyřech datasetech oproti vybraným algoritmům, viz tabulku 4.17, zde je patrné, že klasifikátor KA7-V-V-D vykazuje lepší úspěšnost, než ostatní algoritmy.

5 Ověření robustnosti kontextového klasifikátoru vůči šumu

V této poslední kapitole je podroben analýze robustnosti vůči šumu kontextový klasifikátor s velikostí kontextu 7, s klasickým předzpracováním a se všemi příznaky z inicializačního datasetu, které jsou přidané do kontextového datasetu. Tyto příznaky jsou stejné, jako ty, jenž byly využity pro klasifikátor typu KA7-V-V-D. Je to z toho důvodu, že předzpracování pomocí konvoluční neuronové sítě generuje velmi objemné datasety a výpočty s ním trvají násobně déle a spotřebovávají řádově více operační paměti. Klasifikátor je natrénován na všech trénovacích datech a pak testován na separátních testovacích.

První sada experimentů byla zaměřena na telefonní přenos hlasu, resp. detekci hlasivkových pulsů v řečovém signálu po přenosu telefonní linkou. K tomuto účelu byl vygenerován trénovací a testovací dataset, pro který byl signál speciálně předzpracován, pro simulaci přenosu hlasu telefonní linkou. Nejprve byl na načtený hlasový signál aplikován filtr typu dolní propusť se zlomovou frekvencí 3400 Hz. Následně došlo k podvzorkování signálu na 8 kHz a poté na aplikaci filtru typu horní propusť se zlomovou frekvencí 300 Hz. Následovalo zpětné nadvzorkování na 16 kHz [3]. Zpětné nadvzorkování bylo provedeno pro porovnatelnost výsledků s dalšími experimenty provedenými v této práci. Poté byl proveden výpočet příznaků tak, jak je popsán v podkapitole 4.1. Pro druhou sadu experimentů byl ještě signál před průchodem simulovaným telefonním kanálem zašuměn bílým šumem. Zašumělý signál, po průchodu simulovanou telefonní linkou, měl poměr signál/šum (angl. Signal to Noise Ratio, SNR) -16 dB.

Klasifikátor byl vždy testován na testovacích datech, kde všechna prošla simulovanou telefonní linkou. Výsledky jsou shrnuty v tabulce 5.1. První experiment byl proveden pro trénovací dataset, jehož každá promluva prošla simulovanou telefonní linkou, klasifikátor **KA7-V-V-D-AL1**. Druhý experiment byl proveden pro trénovací dataset 50 na 50, čili půl dat prošlo telefonní linkou a půl ne, klasifikátor **KA7-V-V-D-501**. Třetí experiment byl proveden pro dataset, kde byl spojen dohromady celý dataset, vygenerovaný na čistých datech, s datasetem, jehož data prošla simulovanou telefonní linkou, klasifikátor **KA7-V-V-D-A1**, a poslední experiment používal trénovací da-

taset, vytvořený z čitých dat, klasifikátor **KA7-V-V-D-C1**.

Tabulka 5.1: Porovnání úspěšnosti klasifikátorů pro různé trénovací datasety s různou úrovní dat předzpracovaných simulovanou telefonní linkou.

Metrika	**KA7-V-V-D-C1**	**KA7-V-V-D-A1**	**KA7-V-V-D-501**	**KA7-V-V-D-AL1**
F1	74,595	96,482	96,084	**96,734**
Precision	71,997	96,468	95,713	**96,747**
Avg. prec.	78,007	99,441	99,169	**99,504**
Recall	77,387	96,495	96,458	**96,720**
Accuracy	80,764	97,432	97,131	**97,616**
Bal. acu.	80,046	97,233	96,988	**97,426**
ROC AUC	87,961	99,699	99,565	**99,728**
Brier score	0,185	0,019	0,023	**0,018**

Následně byl pro dva nejlepší případy, tedy pro klasifikátor s trénovacím datasetem spojeným z upraveného a čistého (KA7-V-V-D-A1) a pro klasifikátor s celým upraveným trénovacím datasetem (KA7-V-V-D-AL1), vypočten McNemarův test, viz tabulku 5.2. Nulová hypotéza pro tento test byla, že úspěšnosti správné klasifikace obou klasifikátorů jsou podobné. Alternativní je, že mezi nimi existuje signifikantní rozdíl.

Tabulka 5.2: Kontingenční tabulka pro McNemarův test pro první sadu experimentů s telefonní linkou.

	Úspěch KA7-V-V-D-AL1	**Selhání KA7-V-V-D-AL1**
Úspěch KA7-V-V-D-A1	28389	102
Selhání KA7-V-V-D-A1	156	589

$$\chi^2 = 10{,}888$$

$$p = 0{,}001$$

Druhá sada experimentů byla provedena obdobně, jako v předchozím odstavci, pouze došlo k zašumění dat před aplikováním simulované telefonní linky. Výsledky jsou zapsány v následující tabulce 5.3. Interpretace značení klasifikátorů jsou stejné, pouze přibyl index 2 na konci názvu, značící druhý způsob úpravy dat.

Tabulka 5.3: Porovnání úspěšnosti klasifikátorů pro různé trénovací datasety s různou úrovní dat předzpracovaných simulovanou telefonní linkou a zašumělých daným bílým šumem.

Metrika	**KA7-V-V-D-C2**	**KA7-V-V-D-A2**	**KA7-V-V-D-502**	**KA7-V-V-D-AL2**
F1	71,113	95,742	93,876	**95,962**
Precision	66,318	**96,282**	94,218	96,183
Avg. prec.	71,265	99,263	98,367	**99,297**
Recall	76,655	95,207	93,538	**95,741**
Accuracy	79,587	97,224	96,001	**97,359**
Bal. acu.	78,836	96,707	95,369	**96,944**
ROC AUC	80,683	99,641	99,154	**99,657**
Brier score	0,191	0,021	0,031	0,021

Opět obdobně, jako bylo popsáno výše, byl proveden i McNemarův test pro druhou sadu experimentů, čili pro pro dva nejlepší případy, tedy pro klasifikátor s trénovacím datasetem spojeným z upraveného a čistého (KA7-V-V-D-A2) a pro klasifikátor s celým upraveným trénovacím datasetem (KA7-V-V-D-AL2), viz tabulku 5.2.

Tabulka 5.4: Kontingenční tabulka pro McNemarův test pro druhou sadu experimentů s telefoní linkou a šumem.

	Úspěch KA7-V-V-D-AL2	**Selhání KA7-V-V-D-AL2**
Úspěch KA7-V-V-D-A2	31501	125
Selhání KA7-V-V-D-A2	169	734

$$\chi^2 = 6{,}289$$
$$p = 0{,}012$$

5.1 Diskuze výsledků

Poslední experimenty si kladly za cíl ověřit kontextový klasifikátor KA7-V-V-D proti zašumělým datům. Byl zvolen jen tento typ klasifikátoru z důvodu výpočetní náročnosti pro klasifikátor KAn-C2-V-D a také kvůli skutečnosti, že oba klasifikátory budou kvůli stejné struktuře vykazovat podobné chování pro zašumělá data. Výsledky pro oba typy úpravy dat, použité v těchto experimentech, tedy pro simulovanou telefonní linku a pro zašumělou simulovanou telefonní linku, jsou vypsané v tabulkách 5.1 a 5.3. Z těchto výsledků je patrné, že pro všechna data v trénovacím datasetu, která jsou stejně upravena, je dosaženo nejlepších výsledků. Upravena byla dvěma způsoby, a sice simulovanou telefonní linkou, nebo prošla stejnou linkou a byla předtím i zašuměná bílým šumem. V případě, že podíl upravených dat v datasetu klesá, klesá i úspěšnost klasifikace. Testovací data jsou uvažována všechna upravena. V případě, že neznáme podíl zašumělých dat, nebo jinak upravených, v testovacím datasetu, je rozumné volit konfiguraci se spojením datasetů, kdy spojíme čistý a upravený trénovací dataset. V další práci by bylo vhodné ověřit hypotézu, že při znalosti distribuce zašumělých dat v testovacím datasetu je nejlepší zvolit stejný poměr i v datasetu trénovacím. Závěrem byly provedeny McNemarovy testy pro srovnání konfigurace spojených datasetů a všech trénovacích dat upravených. Ty jsou shrnuté v tabulce 5.2 resp. 5.4. Z nich je patrné, že na hladině významnosti $\alpha = 0{,}05$ je lepší,

pro případ celého zašumělého, nebo upraveného testovacího datasetu, sestavit trénovací dataset rovněž celý zašumělý, či jinak upravený.

6 Závěr

Předkládaná diplomová práce se zabývá detekcí hlasivkových pulsů v řečovém signálu. V první části byl stručně představen fyziologický proces vytváření řeči, převod řečového signálu na signál elektrický, jeho uchování a zpracování v počítači a význam přesné detekce hlasivkových pulsů pro další metody zpracování řečového signálu. Představen byl také použitý klasifikátor XGBoost a způsoby vyhodnocení jeho úspěšnosti. V další části byl uveden výchozí algoritmus pro detekci hlasivkových pulsů vyvíjený na Katedře kybernetiky. Byly zde představeny jednotlivé příznaky a metody předzpracování. Tento algoritmus sloužil jako základ pro následující práci.

Jako další byly představeny nové přidané příznaky, se kterými byl následně natrénován nový klasifikátor, který vykazoval vyšší úspěšnost při detekci hlasivkových pulsů. Na tento krok navazovaly experimenty s novými způsoby předzpracování signálu. Konkrétně se jednalo o předzpracování pomocí vlnkové transformace a použití MS signálu z algoritmu SEDREAMS. Následně byl na rozšířenou množinu příznaků aplikován algoritmus rekurzivní eliminace příznaků, který z rozšířené množiny příznaků vybral optimální podmnožinu. Tato podmnožina byla následně dekorelována, což znamená, že byly vyřazeny ty příznaky, které mezi sebou vykazovaly velkou míru korelace. Poté byl proveden experiment s nalezením jiných, lepších hodnot hyperparametrů klasifikátoru. Po experimentálním ověření vybrané podmnožiny příznaků bylo zjištěno, že tento nový algoritmus je statisticky významně úspěšnější, než algoritmus výchozí.

V dalším kroku byl tento nejlepší nový algoritmus porovnán s novým typem předzpracování pomocí konvoluční neuronové sítě (CNN). Algoritmus s předzpracováním pomocí CNN a velikostí vektorů příznaků 6656 se ukázal jako statisticky významně lepší než nový nejlepší algoritmus. Následně byl implementován kontextový klasifikátor. Ten byl ověřen pro různé velikosti kontextu a pro oba druhy předzpracování, jednak klasické a jednak s pomocí CNN. Experimentálně byla nalezena nejvhodnější velikost kontextu, a sice velikost 7 pro klasické předzpracování, a velikost 6 pro CNN předzpracování. Bylo ukázáno, že tento kontextový klasifikátor je statisticky významně lepší pro klasický typ předzpracování, než nekontextová verze. U předzpracování typu CNN se totéž prokázat napodařilo. Klasifikátor KA7-V-V-D byl poté ověřen vůči vybraným algoritmům na čtyřech testovacích datasetech, viz tabulku 4.17, tento klasifikátor vykazuje větší úspěšnost, než ostatní klasifikátory.

V závěrečné fázi byl nejlepší kontextový klasifikátor ověřen na datech transformovaných simulovanou telefonní linkou a na datech zašumělých bílým šumem a následně transformovaných simulovanou telefonní linkou. Na těchto případech byl ukázán vliv složení trénovacích dat na úspěšnost klasifikátoru.

Za výsledný nejlepší algoritmus lze tedy označit kontextový klasifikátor s kontextem 6 a předzpracováním pomocí CNN s velikostí vektorů příznaků 6656. Při nevhodnosti využití tohoto klasifikátoru z důvodu vysoké výpočetní náročnosti, případně kvůli neinterpretovatelnosti příznaků, lze doporučit kontextový klasifikátor s kontextem velikosti 7 a s klasickým předzpracováním. V případě, že klasifikátor bude pracovat na datech, která jsou zašuměná, či jinak upravená, je vhodné stejným způsobem upravit i trénovací data, tím bude dosaženo nejlepší úspěšnosti klasifikace.

Seznam zkratek

- AD - Analog to discrete - Analogový na diskrétní
- DA - Discrete to analog - Diskrétní na analogový
- CART - Classification and regresion tress - Klasifikační a regresní stromy
- EEG - Elektroglotograf - Elektroglotograf
- FT - Fourier Transform - Fourierova transformace
- FFT - Fast Fourier Transform - Rychlá Fourierova transformace
- FLAC - Free Lossless Audio Codec - Volný bezztrátový kodek audia
- GCI - Glottal closure instant - Hlasivkový puls
- MS - Mean-based signal - Signál založený na střední hodnotě
- OS - Operating system - Operační systém
- RIFF - Resource interchange file format - Formmát souborů pro výměnu zdrojů
- ROC AUC - Reciever Operatincg Characteristics Area Under Curve - Oblast pod křivkou operační charakteristiky přijímače
- STFT - Short time Fourier transform - krátkodobá Fourierova transformace
- TD-PSOLA - Time Domain Pitch Synchronous Overlap and Add - metoda přesahu a přidání se sychroním zpracováním hlasivkových pulsů v časové oblasti
- WAV - Waveform audio file format - Formát pro uchovávání audio nahrávek
- ZFF - Zero Frequency Filtering - Filtrování nulové frekvence

Seznam obrázků

Seznam tabulek

Literatura

[1] *FestVox Speech Synthesis Databases* [online]. [Citováno 24. 5. 2020]. Dostupné z: `http://festvox.org/dbs/index.html`.

[2] Achuth, R. M. – Ghosh, P. K. PSFM - A Probabilistic Source Filter Model for Noise Robust Glottal Closure Instant Detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*. 2018, 26, 9, s. 1645–1657.

[3] Andersen, B. et al. *Telephone filter* [online]. [Citováno 24. 5. 2020]. Dostupné z: `http://kom.aau.dk/group/04gr742/pdf/telefilter_worksheet.pdf`.

[4] Brownlee, J. *XGBoost With Python: Gradient Boosted Trees with XGBoost and scikit-learn*. Machine Learning Mastery, 2016.

[5] Buitinck, L. et al. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, s. 108–122, 2013.

[6] Chen, T. – Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, s. 785–794, New York, NY, USA, 2016. ACM.

[7] Chunli, W. – Chunlei, Z. – pengtu, Z. Denoising algorithm based on wavelet adaptive threshold. *Physics Procedia*. 12 2012, 24, s. 678–685.

[8] Coalson, J. *FLAC - features* [online]. Xiph.Org Foundation. [Citováno 24. 5. 2020]. Dostupné z: `https://xiph.org/flac/features.html`.

[9] Colotte, V. – Laprie, Y. Higher precision pitch marking for TD-PSOLA. In *2002 11th European Signal Processing Conference*, s. 1–4, 2002.

[10] Creative technology ltd. *Sound Blaster Series Hardware Programming Guide* [online]. [Citováno 24. 5. 2020]. Dostupné z: `http://www.scs.stanford.edu/05au-cs240c/lab/hardware/SoundBlaster.pdf`.

[11] Degottex, G. et al. COVAREP - A collaborative voice analysis repostiory for speech technologies. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014.

[12] DRUGMAN, T. et al. Detection of glottal closure instants from speech signals: A quantitative review. *IEEE Transactions on Audio, Speech, and Language Processing*. 2012, 20, s. 994–1006.

[13] EATON, J. W. et al. *GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations* [online]. [Citováno 24. 5. 2020]. Dostupné z: `https://www.gnu.org/software/octave/doc/v5.2.0/`.

[14] FRIEDMAN, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*. 2000, 29, s. 1189–1232.

[15] GARCIA-GASULLA, D. et al. On the Behavior of Convolutional Nets for Feature Extraction. *CoRR*. 2017, abs/1703.01127.

[16] GIANNAKOPOULOS, T. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. *PloS one*. 2015, 10, 12.

[17] HERBST, C. T. et al. Glottal opening and closing events investigated by electroglottography and super-high-speed video recordings. *Journal of Experimental Biology*. 2014, 217, 6, s. 955–963.

[18] HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, 9, 3, s. 90–95.

[19] JAN, J. *Číslicová filtrace, analýza a restaurace signálů*. Vysoké učení technické Brno, nakladatelství VITUM, 2 edition, 2002.

[20] KABAL, P. *Audio File Format Specifications* [online]. MMSP Lab, ECE, McGill University. [Citováno 24. 5. 2020]. Dostupné z: `http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html`.

[21] KADIRI, S. R. A Quantitative Comparison of Epoch Extraction Algorithms for Telephone Speech. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, s. 6500–6504, 2019.

[22] KANE, J. *Tools for analysing the voice : developments in glottal source and quality analysis*. PhD thesis, Trinity College. Centre for Language and Communication Studies, Dublin, Ireland, 2012.

[23] KANE, J. – GOBL, C. Identifying Regions of Non-Modal Phonation Using Features of the Wavelet Transform. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, s. 177–180, 01 2011.

[24] Khanagha, V. – Daoudi, K. – Yahia, H. M. Detection of Glottal Closure Instants Based on the Microcanonical Multiscale Formalism. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* December 2014, 22, 12, s. 1941–1950.

[25] Kinnunen, T. – Li, H. An Overview of Text-Independent Speaker Recognition: from Features to Supervectors. *Speech Communication*. 01 2010, 52, s. 12–40.

[26] Klapuri, A. – Davy, M. Signal processing methods for music transcription, chapter 5. *Springer Science and Business Media*. 2007.

[27] Kodukula, S. R. M. – Yegnanarayana, B. Epoch Extraction From Speech Signals. *Audio, Speech, and Language Processing, IEEE Transactions on*. 12 2008, 16, s. 1602 – 1613.

[28] Koutrouvelis, A. I. et al. A fast method for high-resolution voiced/unvoiced detection and glottal closure/opening instant estimation of speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2016, 24, 2, s. 316–328.

[29] Krčmová, M. *Kapitola 5 Artikulační (organogenetická, fyziologická) fonetika* [online]. Filozofická fakulta Masarykovy univerzity. [Citováno 24. 5. 2020]. Dostupné z: `https://is.muni.cz/do/1499/el/estud/ff/js08/fonetika/ucebnice/index.html`.

[30] Krčmová, M. *Úvod do fonetiky a fonologie pro bohemisty*. Ostravská univerzita, Filozofická fakulta, 1 edition, 2006.

[31] Krzywinski, M. – Altman, N. Visualizing samples with box plots. *Nat Methods*. 02 2014, 11, s. 119–20.

[32] Labutin, P. – Koval, S. – Raev, A. Speaker identification based on the statistical analysis of f0. 01 2007.

[33] Laprie, Y. – Colotte, V. Automatic pitch marking for speech transformations via TD-PSOLA. In *9th European Signal Processing Conference (EUSIPCO 1998)*, s. 1–4, 1998.

[34] Lee, G. et al. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*. 2019, 4, 36, s. 1237.

[35] Legát, M. – Matoušek, J. – Tihelka, D. A robust multi-phase pitch-mark detection algorithm. In *INTERSPEECH*, 1, s. 1641–1644, Antwerp, Belgium, 2007.

[36] Lyons, J. et al. jameslyons/python_speech_features: release v0.6.1. January 2020.

[37] Matoušek, J. – Tihelka, D. Classification-Based Detection of Glottal Closure Instants from Speech Signals. In *International Speech and Communication Association, Interspeech 2017*, Hyderabad, 2017.

[38] Matoušek, J. – Tihelka, D. Glottal Closure Instant Detection from Speech Signal Using Voting Classifier and Recursive Feature Elimination. In *International Speech and Communication Association, Interspeech 2018*, Brighton, United Kingdom, 2018.

[39] Matoušek, J. – Tihelka, D. Using Extreme Gradient Boosting to Detect Glottal Closure Instants in Speech Signal. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, s. 6515–6519, Brighton, United Kingdom, 2019.

[40] Matoušek, J. – Vraštil, M. Context-Aware XGBoost for Glottal Closure InstantDetection in Speech Signal. In *TSD 2020, 23rd International Conference on Text, Speech and Dialogue*, 2020. [Přijato].

[41] McFee, B. et al. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, s. 18–24, 01 2015.

[42] McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*. jun 1947, 12, 2, s. 153–157.

[43] Mišurec, J. *Základní metody číslicového zpracování signálů pro integrovanou výuku VUT a VŠB-TUO*. Vysoké učení technické Brno, 1 edition, 2014.

[44] Moulines, E. – Verhelst, W. Time-Domain and Frequency-Domain Techniques for Prosodic Modification of Speech. *Speech Coding and Synthesis*. 01 1995.

[45] Naylor, P. A. et al. Estimation of glottal closure instants in voiced speech using the DYPSA algorithm. *IEEE Transactions on Audio, Speech and Language Processing*. 2007, 15, 1, s. 34–43.

[46] Naylor, P. A. et al. Estimation of Glottal Closure Instants in Voiced Speech Using the DYPSA Algorithm. *Trans. Audio, Speech and Lang. Proc.* January 2007, 15, 1, s. 34–43.

[47] Oliphant, T. *NumPy: A guide to NumPy* [online]. 2006. [Citováno 24. 5. 2020]. Dostupné z: `http://www.numpy.org/`.

[48] Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, 12, s. 2825–2830.

[49] Psutka, J. et al. *Mluvíme s počítačem česky.* Praha : Academia, 1 edition, 2006.

[50] Raissi, R. *The Theory behind Mp3* [online]. 2002. [Citováno 24. 5. 2020]. Dostupné z: https://www.mp3-tech.org/programmer/docs/mp3_theory.pdf.

[51] Raschka, S. MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *The Journal of Open Source Software.* April 2018, 3, 24.

[52] Red Hat Inc. a další. *Get Fedora* [online]. [Citováno 24. 5. 2020]. Dostupné z: https://getfedora.org/.

[53] Satran, M. *Resource Interchange File Format (RIFF) - Win32 apps* [online]. Microsoft Corporation, May 2018. [Citováno 24. 5. 2020]. Dostupné z: https://docs.microsoft.com/en-us/windows/win32/xaudio2/resource-interchange-file-format--riff-.

[54] Simonyan, K. – Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014.

[55] Talkin, D. *REAPER: Robust Epoch And Pitch EstimatoR* [online]. Google corporation, Dec 2019. [Citováno 24. 5. 2020]. Dostupné z: https://github.com/google/REAPER.

[56] Thomas, M. – Gudnason, J. – Naylor, P. A. Estimation of Glottal Closing and Opening Instants in Voiced Speech using the YAGA Algorithm. *IEEE Trans. Audio, Speech, Lang. Process.* June 2016, 20, s. 82–91.

[57] Van Rossum, G. – Drake, F. L. *Python 3 Reference Manual.* CreateSpace, 2009.

[58] Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods.* 2020, 17, s. 261–272.

[59] Waskom, M. et al. mwaskom/seaborn: v0.10.0 (January 2020). Jan 2020.

[60] WELCH, P. The use of the fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust.* 1967, 15, s. 70–73.

[61] McKinney. Data Structures for Statistical Computing in Python, 2010.

[62] Wirsum, S. *Abeceda nf techniky.* BEN : Technická Literatura, 1998.

Příloha A

Seznam značení pro různé typy použitých klasifikátorů.

- **X-V-V-V** - Klasifikátor XGBoost s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a výchozí množinou příznaků. Představený byl v podkapitole 3.3.
- **X-V-V-N** - Klasifikátor XGBoost s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou množinou příznaků. Představený byl v kapitole 4.
- **X-MS-V-N** - Klasifikátor XGBoost s předzpracováním pomocí Mean-Based signálu z algoritmu SEDREAMS, výchozími hodnotami hyperparametrů a rozšířenou množinou příznaků. Představený byl v podkapitole 4.2.
- **X-W-V-N** - Klasifikátor XGBoost s předzpracováním pomocí vlnkové transformace, výchozími hodnotami hyperparametrů a rozšířenou množinou příznaků. Představený byl v podkapitole 4.2.
- **X-V-V-S** - Klasifikátor XGBoost s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou a vybranou množinou příznaků. Představený byl v podkapitole 4.3.
- **X-V-V-D** - Klasifikátor XGBoost s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.3.
- **X-V-N-D** - Klasifikátor XGBoost s výchozím typem předzpracování, novými hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.4.
- **X-C2-V-D** - Klasifikátor XGBoost s předzpracováním pomocí konvoluční neuronové sítě, s velikostí vektorů příznaků 6656, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.5.
- **X-C1-V-D** - Klasifikátor XGBoost s předzpracováním pomocí konvoluční neuronové sítě s velikostí vektorů příznaků 64, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.5.
- **X-8-V-D** - Klasifikátor XGBoost s výchozím typem předzpracování, avšak s 8 kHz referenčními hodnotami o kandidátech na hlasivkový puls v signálu, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v pod-

kapitole 4.5.

- **Kn-V-V-D** - Kontextový klasifikátor XGBoost s kontextem n, výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.6.
- **KAn-V-V-D** - Kontextový klasifikátor XGBoost s kontextem n a všemi příznaky z inicializačního datasetu, s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.6.
- **KAn-C2-V-D** - Kontextový klasifikátor XGBoost s kontextem n a všemi příznaky z inicializačního datasetu, s předzpracováním pomocí konvoluční neuronové sítě s velikostí vektorů příznaků 6656, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Představený byl v podkapitole 4.6.
- **KA7-V-V-D-C1** - Kontextový klasifikátor XGBoost s kontextem 7 a všemi příznaky z inicializačního datasetu, s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Trénovaný byl na čistém datasetu a testovaný na datasetu, jehož data prošla simulovanou telefonní linkou. Představený byl v kapitole 5.
- **KA7-V-V-D-A1** -Kontextový klasifikátor XGBoost s kontextem 7 a všemi příznaky z inicializačního datasetu, s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Trénovaný byl na spojeném čistém datasetu a datasetu, jehož data prošla simulovanou telefonní linkou. Testovaný pak byl na stejně upraveném datasetu. Představený byl v kapitole 5.
- **KA7-V-V-D-501** - Kontextový klasifikátor XGBoost s kontextem 7 a všemi příznaky z inicializačního datasetu, s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Trénovaný byl na spojeném čistém datasetu a datasetu, jehož data prošla simulovanou telefonní linkou, z obou bylo bráno však jen 50 % dat bez překrytí. Testovaný pak byl na stejně upraveném datasetu. Představený byl v kapitole 5.
- **KA7-V-V-D-AL1** - Kontextový klasifikátor XGBoost s kontextem 7 a všemi příznaky z inicializačního datasetu, s výchozím typem předzpracování, výchozími hodnotami hyperparametrů a rozšířenou, vybranou a dekorelovanou množinou příznaků. Trénovaný byl na datasetu, jehož data prošla simulovanou telefonní linkou. Testovaný pak byl na stejně

upraveném datasetu. Představený byl v kapitole 5.

- **KA7-V-V-D-C2**, **KA7-V-V-D-A2**, **KA7-V-V-D-502**, **KA7-V-V-D-AL2** - Tyto klasifikátory jsou obdobou předchozích s indexem 1 na konci názvu. Rozdíl je v použité úpravě dat. Ta nejen že prošla simulovanou telefonní linkou, ale předtím byla zašuměna bílím šumem tak, aby výsledný signál měl odstup signál šum -16 dB. Tyto klasifikátory byly představeny v kapitole 5.

Příloha B

Výpis architektury použité konvoluční neuronové sítě pro extrakci a selekci příznaků z řeči. Pro trénování byl použit Adadelta optimalizér, mini-batch o velikosti 32 vzorků, dropout 50 %, počet epoch 200, na vstupu byl řečový signál vzorkovaný 8 kHz frekvencí s okny kolem každého kandidáta na hlasivkový puls velikosti 50 ms.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 401, 1)	0
block1_conv1 (Conv1D)	(None, 401, 64)	512
block1_conv2 (Conv1D)	(None, 401, 64)	28736
block1_bn (BatchNormalizatio	(None, 401, 64)	256
block1_pool (MaxPooling1D)	(None, 201, 64)	0
block1_dropout0.5 (Dropout)	(None, 201, 64)	0
block2_conv1 (Conv1D)	(None, 201, 128)	41088
block2_conv2 (Conv1D)	(None, 201, 128)	82048
block2_bn (BatchNormalizatio	(None, 201, 128)	512

```
block2_pool (MaxPooling1D)    (None, 101, 128)
0
_________________________________________________________________
block2_dropout0.5 (Dropout)   (None, 101, 128)
0
_________________________________________________________________
block3_conv1 (Conv1D)         (None, 101, 256)
98560
_________________________________________________________________
block3_conv2 (Conv1D)         (None, 101, 256)
196864
_________________________________________________________________
block3_bn (BatchNormalizatio (None, 101, 256)
1024
_________________________________________________________________
block3_pool (MaxPooling1D)    (None, 51, 256)
0
_________________________________________________________________
block3_dropout0.5 (Dropout)   (None, 51, 256)
0
_________________________________________________________________
block4_conv1 (Conv1D)         (None, 51, 512)
393728
_________________________________________________________________
block4_conv2 (Conv1D)         (None, 51, 512)
786944
_________________________________________________________________
block4_bn (BatchNormalizatio (None, 51, 512)
2048
_________________________________________________________________
block4_pool (MaxPooling1D)    (None, 26, 512)
0
_________________________________________________________________
block4_dropout0.5 (Dropout)   (None, 26, 512)
0
_________________________________________________________________
block5_conv1 (Conv1D)         (None, 26, 512)
786944
_________________________________________________________________
```

```
block5_conv2 (Conv1D)           (None, 26, 512)
786944
_________________________________________________________________
block5_bn (BatchNormalizatio (None, 26, 512)
2048
_________________________________________________________________
block5_pool (MaxPooling1D)    (None, 13, 512)
0
_________________________________________________________________
block5_dropout0.5 (Dropout)  (None, 13, 512)
0
_________________________________________________________________
flatten_1 (Flatten)             (None, 6656)
0
_________________________________________________________________
fc1 (Dense)                     (None, 64)
426048
_________________________________________________________________
fc_bn (BatchNormalization)    (None, 64)
256
_________________________________________________________________
fc_dropout0.5 (Dropout)       (None, 64)
0
_________________________________________________________________
predictions (Dense)             (None, 1)
65
=================================================================
Total params: 3,634,625
Trainable params: 3,631,553
Non-trainable params: 3,072
_________________________________________________________________
```