České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra elektromagnetického pole

Detekce podvržené řeči

SEMESTRÁLNÍ PROJEKT

Autor: Bc. Jiří Šmíd
Vedoucí: Doc. Ing. Petr Pollák, CSc.
Datum: Květen, 2025

Seznam použitých zkratek

ASR	Automatické rozpoznání řečníka (z anglického *Automatic Speaker Verification*)
ASV	Automatické ověření řečníka (z anglického *Automatic Speaker Verification*)
CNN	Konvoluční neuronová síť (z anglického *Convolutional Neural Network*)
DNN	Hluboká neuronová síť (z anglického *Deep Neural Network*)
GMM	Model gausovských směsí (z anglického *Gaussian Mixture Model*)
LA	Virtuální přístup (z anglického *Logical Access*)
LFCC	Lineárně frekvenční kepstrální koeficienty (z anglického *Linear Frequency Cepstral Coeficients*)
LR	Rychlost učení (Z anglického *Learning Rate*)
LSTM	Dlouhá krátkodobá paměť (z anglického *Long Short Time Memory*).
MFCC	Mel-frekvenční kepstrální koeficienty (z anglického *Mel-frequecy Cepstral Coeficients*)
PA	Fyzický přístup (z anglického *Physical access*)
ResNet	Residuální neuronová síť (z anglického *Residual Neural Network*)
NN	Neuronová síť (z anglického *Neural Network*)
RNN	Rekurentní neuronová síť (z anglického *Recurent Neural Network*)
TTS	Převodník textu na řeč (z anglického *Text to Speech*)
VC	Hlasová konverze (z anglického *Voice Conversion*)

Obsah

1 Úvod

Historie strojově generované řeči sahá až do 20. let minulého století, kdy byly představeny první elektronické obvody schopné generovat jednotlivé hlásky, později i slova. Využitelná uměle vytvořená řeč se však objevuje až s rozvojem počítačových technologií a statistického modelování řeči. Průlomem v této technologii byly skryté markovovy modely a v nedávné době neuronové sítě.

Dnes využívané nástroje pro generování, či konverzi řeči dokáží uživatelům zprostředkovávat službu bez výrazných (slyšitelných) artefaktů, vygenerovaná promluva není monotónní a již se neobjevuje typická trhanost dřívějších systémů. Tyto nástroje dnes nachází široké využití: na telefoních infolinkách, v domácích asistentech, pro namlouvání voiceoverů videí na sociálních sítích, pro pomoc lidem se zrakovým postižením a pro mnoho dalších.

S rozvojem těchto technologií, jejich dostupností a relativní jednoduchostí jejich používání narostla šance jejich zneužití. Mnoho z nástrojů generování řeči, či její konverze z řečníka na řečníka dokáže i z poměrně malého záznamu promluvy člověka jeho hlas poměrně důvěryhodně napodobit. Toho může být zneužito pro různé druhy podvodů, ovlivňování veřejnosti, či pro útoky na biometrické systémy, které hlas využívají k identifikaci.

Cílem tohoto projektu je seznámit se současnými metodami detekce podvržené řeči, nalezení vhodného datasetu, který obsahuje přirozené i uměle generované promluvy, na kterých půjdou tyto metody natrénovat a ověřit. Dalším z kroků je pak implementace vybraných, či vlastních řešení a ověřit jejich konkurenceschopnost na dostupných datech. Všechny tyto kroky slouží jako příprava pro diplomovou práci, jejímž úkole bude vytvoření takovéhoto detektoru pro český jazyk, pro který žádné volně dostupné datasety neexistují. Takovýto detektor, v době psaní práce, není pro češtinu volně dostupný, ani není dostupný pro akademickou sféru. O privátních a neveřejných nástrojích nejsou k dohledání informace.

2 Strojově generovaná řeč

2.1 Stručná historie strojově generované řeči

První pokusy o vytvoření umělé řeči resp. umělých hlásek se objevují v 18. století, kdy byly popsány fyziologické rozdíly mezi pěti základními samohláskami. První pokusy se skládaly z pěti rezonátorů, jeden pro každou samohlásku. Později se objevují mechanicko-akustické syntetizátory, které jsou schopné již generovat i souhlásky, včetně nasálních i explozivních.

S rozvojem elektrických obvodů ve 20. století přichází na scénu elektronické generátory řeči. Podobně jako u mechanických generátorů jsou z počátku schopny generovat základní samohlásky (pouze se dvěma formanty), později však i samohlásky. Prvním systémem, který lze považovat za generátor řeči je VODER (Voice Operating Demonstrator), představený roku 1939. Výstupy tohoto nástroje byly poměrně vzdáleny od přirozené řeči, byl však prokázán potenciál pro strojové generování řeči.

V druhé polovině 20. století jsou pak vyvinuty různé pokročilejší syntezátory založeny na formantové syntéze, kaskádní formantové syntéze a modelování přenosové funkce vokálního traktu pro různé hlásky. Později se přidává i modelování artikulace. První kompletní TTS systém byl představen v roce 1969 a zahrnoval artikulační model i syntaktickou analýzu s pokročilými heuristikami. Později se objevují systémy, které již nejsou založeny pouze na několika parametrech popisující jednotlivé hlásky (fóny). Systémy již využívají databáze difónů (druhá polovina aktuálního fónu a první polovina následujícího), které jsou řazeny za sebe podle zadané sekvence znaků a převedeny na řečový signál např. pomocí LP analýzy [1].

První počítačové modely byly taktéž založeny primárně na formantové syntéze a nastavování přenosových funkcí vokálního traktu (VOCODER). S rozvojem výpočetního výkonu a pamětí počítačů, byly nahrány velké datasety a použity pro výběr parametrů, které lépe popisují fonémový i lingvistický kontext a slouží pro generování řeči s lepší prosodií, důrazy atd. Tyto systémy obsahovaly mnoho parametrů a byly složité na nastavení, proto vzniká nový způsob popisu a tj. statistická parametrická syntéza řeči, která využívá statistiky získané z nahraných promluv. Průlomem v tomto odvětví byly HMM (skryté Markovovy modely), které nepopisují pouze sekvence fonémů, ale i mnoho dalších kontextových informací. S rozvojem technologií strojového učení a metod jako Viterbiho algoritmus, či Baum-Welch se na dlouho dobu staly HMM nejpoužívanějším přístupem [2].

V posledních letech, kdy techniky neuronových sítí a strojového učení pokročily, je v popředí syntéza řeči za pomocí nástrojů strojového učení. Oproti tradičním

přístupům, nabízejí tyto nástroje řadu výhod: jednodušší adaptace na cílového řečníka, možnost vyjádření emocí, přirozená prosodie a další. Toto je způsobeno schopností neuronových sítí naučit se a využít komplexní vzory a charakteristiky lidské řeči.

2.2 Syntéza řeči

Tato sekce se zabývá stručným popisem základních principiálních struktur pro syntézu řeči tradičními nástroji i E2E systémy založenými na NN. Podrobnější popis systémů není pro tuto práci relevantní.

2.2.1 Tradiční systémy

Tradiční systémy systémy syntézy řeči se obvykle skládají z několika modulů sériově řazených.

První z modulů obvykle převádí text na fóny s prosodickými značkami (bohatá fonetická transkripce). Skládá se ze tří bloků: analyzátor textu, fonetický analyzátor a generátor prosodie. Analyzátor textu zajišťuje preprocessing, morfologickou,kontextovou, syntaktickou a prosodickou analýzu (k těm je nutno dodat další nástroje/soubory s gramatikou atd.)

Další z modulů převádí fonetickou transkripci na akustický signál resp. na příznakovou reprezentaci (spectrogram, mel-spectrogram ...). Jako vstup taktéž potřebuje informace o řečníkovi tj. informace o rychlosti, intonaci, důrazech aj. Jednodušší z těchto metod je formantová syntéza. Tanto syntéza vychází ze zjednodušeného modelu vokálního traktu. Buzení systému odpovídá vibracím/nevybracím hlasivek (pulsy, šum), systém sám pak modeluje hlasový trakt (přenosové funkce - formanty). Jedním z používaných systémů je konkatenační analýza. Ta skládá výslednou řeč z předem nahraných úseků řeči, které se spojují za sebe. Tento typ syntézy vyžaduje rozsáhlou databázi nahraných řečových segmentů pokrývající všechny možné kombinace zvuků v daném jazyce.

Jako poslední může následovat blok, který vygenerované příznaky převádí na řečový signál, pokud již není generována přímo systémem.

Tradiční systémy jsou struktury, které vyžadují poměrně velké množství kroků a potřebují mnoho externích databází, či sad pravidel. Je proto poměrně složité takovýto nástroj vytvořit. U těchto systémů je také složité změnit výstupní charakteristiky řeči tj. změnit řečníka. Další nevýhodou je jazyková nepřenositelnost tj. model potřebuje pro každý jazyk jiné sady pravidel a databáze [3].

2.2.2 Systémy založené na ML a NN

Systémy založené na NN mohou být E2E, či se mohou skládat z několika částí. Některé z nich umožňují pouze TTS, jiné i převod textu na řeč (SST - z anglického *Speech To Text*) či převod řeči na řeč (hlasová konverze - převod řeči z jednoho řečníka na jiného, VC - z anglického *Voice Conversion*). Jejich hlavní výhoda spočívá v tom, že ke svému učení a běhu nepotřebují sady pravidel pro daný jazyk. Jistou reprezentaci těchto pravidel (statistický pravděpodobnostní popis) si struktura vybuduje během učení, kdy je na vstup předkládán text (či jeho vektorizovaná podoba) a jako výstup (label) pak odpovídající řečový systém. Níže jsou uvedeny nejpoužívanější nástroje:

Speech T5

Tento model vyvinutý Microsoft Research v sobě kombinuje tři úlohy: TTS, STT a VC a je založen na architektuře transformerů. Jádro je tvořené encoder-decoder strukturou a k dispozici je šest pomocných sítí (preprocess, postprocess) pro konkrétní aplikace [4]. Pro TTS je vstupem modelu vektorizovaný text pomocí SpeechT5Processor. Výstupem je pak mel-spektrogram, který lze převést na řečový signál pomocí WaveGlow [5]. Model byl trénován na anglických datasetch, ale lze jej přetrénovat či dotrénovat i pro jiné jazyky. Na Hugging Face jsou k dispozici dvě české mutace. První je kompletně natrénované jádro na českých datasetch [6], druhým pak je model s anglickým jádrem dotrénovaný na české části VoxPopuli datasetu [7],[8]. Experimenty provedené autory prvního z uvedených modelů ukázaly, že takovýto model je schopen velmi dobře rekonstruovat řečníka i z přibližně minuty hlasového záznamu (speaker embeding) [9], [10].

Tacotron 2

Jedná se o autoregresní model vyvinutý Googlem. Podobně jako SpeechT5, nepotřebuje promluvu rozloženou na jednotlivé fonémy, ale postačí mu prostý přepis promluvy (u některých implementací možný i fonémový přepis). Obsahuje mechanismus pozornosti (attention mechanism), který umožňuje lepší synchronicitu mezi textem a výstupem, kterým je mel-spektrogram. Tacotron 2 standardně používá Wave Glow pro syntézi signálů, ale lze využít i jiný systém. Stejně jako předchozí systém i Tacotron umožňuje vložení externích řečových charakteristik cílového řečníka i přenos hlasu řečníka jiného jazyka na jazyk na který byl systém trénován [11].

FastSpeech

Systém, který vznikl za účelem vyřešení problémů s autoregresivními modely jako Tacotron 2, které jednotlivé segmenty výstupního spektrogramu generují sekvenčně.

To přináší nevýhody jako: delší čas generování, možnost kaskádní propagace chyby či občasné problémy se synchronizací (selhání pozornostních mechanismů). Vstupem je text a výstupem mel-spektrogram [12].

Kromě zde uvedených modelů existují a jsou hojně užívány např. Glow-TTS a VITS.

2.3 Využití strojově generované řeči

Rozvoj nástrojů a technik strojového učení a umělé inteligence přinesl nejen v oblasti zpracování řeči, ale i např. zpracování obrazu, doposavad neuskutečnitelné možnosti, a to i v oblastech syntézy a úpravy. Tyto moderní nástroje jsou schopny již velmi věrně generovat řeč, která je téměř nerozlišitelná od přirozené řeči či se podobností lidské řeči blíží a není pro člověka nepříjemné ji poslouchat. Tohoto bývá využíváno v mnoha aplikacích, obecně především v komunikaci mezi strojem a člověkem. Mezi tyto aplikace patří: předčítání textu či obsahu webových stránek pro osoby se zrakovým postižením [13], komunikace zařízení chytrých domácností a mobilních asistentů (Google Home, Alexa aj.), nebo také komunikace hlasových chatbotů, jejichž úkolem je získat informace od zákazníka či mu je poskytnout, než bude spojen s lidským operátorem. Tohoto je poměrně často využíváno v bankách a podobných institucích [14]. Další z častých využití jsou automatické hlásící systémy např. na nádražích či letištích.

Mezi další aplikace v poslední době, také patří vytváření obsahu, který je pak zveřejňován v televizi, či na sociálních sítích na internetu, jelikož tyto systémy nabízejí jednodušší, rychlejší a levnější způsob, oproti namlouvání textu lidmi. Další z výhod je možnost mnoha jazykových mutací daného obsahu.

2.4 Zneužití strojově generované či pozměněné řeči

Možnost strojově generovat, či pozměnit řeč, tak aby zněla jako řeč konkrétní osoby s sebou přinesla i možnost zneužití. Tato možnost zneužití ještě vzrostla s příchodem systémů, které pro adaptaci na řečníka nepotřebují dlouhý záznam jeho promluv, ale pouze krátkou promluvu [15] [16]. Existují v zásadě tři druhy zneužití.

Prvním z nich je cílený telefonní/videotelefonní útok na konkrétní osobu, jehož účelem je oklamání oběti vydáváním se za někoho jiného, blízkého (kamarád, nadřízený, člen rodiny) a přimět ho něco vykonat. Druhým z nich je pak vytvoření deepfaku, nejčastěji i s obrazovou částí, obvykle známé a vlivné osobnosti, který je následně zveřejněn za účelem poškození dané osoby, ovlivnění veřejného mínění či dodání zdání legitimity a pravosti nějakého podvodu. Třetí se sestává z útoku na systémy automatického ověření řečníka (ASV - z Anglického *Automatic Speaker Verification*) v bankách i jiných institucích, kde bývají jisté příznaky řeči využity pro biometrické ověření klienta.

Pro první dva typy útoků bývá typické, že neobsahují pouze podvrženou řeč. Mnohdy bývá přítomno i podvržené video, podvodné emaily a další typy podvodného jednání. Při útocích na společnost mohou být přítomny i botnety sloužící k amplifikaci narativu.

2.4.1 Útoky cílené na konkrétní osobu

Tento typ útoků cílí obvykle na jednotlivce, nebo firmy obvykle s cílem vylákání peněz. Útočníci se nejčastěji vydávají za vysoké představitele firem a požadují po svých "podřízených, či partnerech"převod peněz.

Toto dokládá případ z roku 2019 z Anglie, kdy útočníci uskutečnili telefonát s CEO jedné energetické firmy a vydávali se za CEO její mateřské firmy. Hlas útočníka byl pozměněn/generován tak, aby zněl jako jeho hlas. Útočníkům se podařilo z firmy vylákat 243 000 USD [17].

Další případ se odehrál v roce 2020 v Hong Kongu. Zde se útočníci zavolali manažerovi jedné japonské firmy. Hlas v telefonu zněl jako hlas ředitele mateřské firmy. "Ředitel"informoval manažera o nových akvizicích firmy a nutnosti přeposlat 35 000 000 USD. Pro dodání věrohodnosti byly součástí útoku i podvodné emaily. Z odeslaných peněz se podařilo vystopovat asi jen 400 000 USD [18].

Ne vždy jsou útoky úspěšné. Jedním z těchto neúspěšných pokusů je i případ zneužití identity CEO jedné největších reklamních agentur WPP Marka Reada. V tomto případě útočníci založili WhatsApp účet, který vypadal jako Readův. Z něj následně sjednali online setkání na Microsoft Teams s vedoucími představiteli firmy, ohledně nových obchodních příležitostí, za účelem odcizení peněz a osobních údajů. V tomto útoku sehrál důležitou roli nástroj hlasové konverze (VC - z anglického *Voice Convertion*) [19].

2.4.2 Podvržená řeč známých a vlivných osob

Prvním z důvodů takovéhoto podvodu bývá poškození dobrého jména exponované osoby. Motivace v tomto případě může být osobní, politická, potenciálně i ekonomická. Jedním z příkladů politické motivace je podvržená nahrávka, ve které se údajný předseda strany Progresívne Slovensko Michal Šimečka baví s údajnou novinářkou Monikou Tódovou a řeší spolu, jak ovlivní volby [20][21].

Dalším obětí tohoto typu podvrhu se stal Keir Starmer (v té době předseda Labouristů). Podvodná nahrávka byla zveřejněna první den sjezdu strany v Liverpoolu.

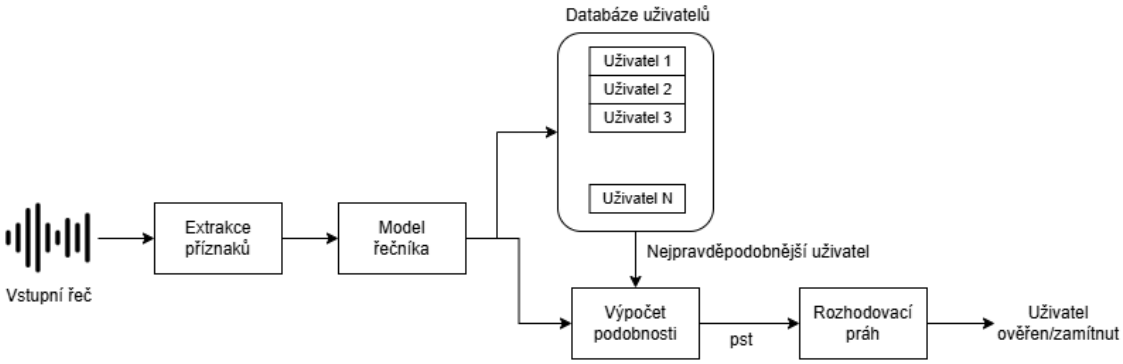
Na této podvodné nahrávce je zachyceno (vytvořeno), jak Starmer ponižuje svoje zaměstnance a vyjadřuje se velmi kriticky k Liverpoolu, kde se sjezd konal.Nahrávka byla přehrána více než milionkrát [22].

Dalším z důvodů může být snaha dodat zdání důvěryhodnosti nějakému podvodnému projektu, cílícímu na občany. Jedním z příkladů může být zneužití identity moderátorek CNN Prima News Pavlíny Wolfové a Kateřiny Hošek Štruncové či reportérů Jakuba Kokoška a Ondřeje Němce. V podvržených (deepfake) videích propagují tito novináři různé podvodné investiční projekty slibující veliké zisky za minimální peníze [23]. Dalším z příkladů může být zneužití identity UFC zápasníka Jiřího Procházky. V tomto podvodném videu nabízí údajný Procházka 50 mil. korun přes "svojí"sázkovou aplikaci Perfect Game [24]. V obou případech hrála podvržená řeč zásadní roli. Podvod cílený na mnohem větší množství lidí zneužil identitu youtubera MrBeasta, tvůrce cílícího především na mladší diváky (děti a teenageři). V tomto podvodném klipu nabízí údajný MrBeast rozdávání nových Iphonů 15 pouze za 2 dolary [25].

Kvalita videí u takovýchto podvodů mnohdy nebývá valná a člověk znalý velice rychle odhalí, že s videem je něco v nepořádku. Audio bývá často v lepší kvalitě, ale taktéž jsou mnohdy slyšitelné artefakty: řeč nemá přirozené tempo, je monotónější atd. Přes toto všechno si podvody najdou svůj cíl a mnoho lidí je není schopno rozlišit resp. ani nepřemýšlí, že by se mohlo jednat o podvod. Důvody, proč tomu tak je a jak to zlepšit jsou však tématem pro jiný obor.

2.4.3 Útoky cílené na ASV systémy

ASV jsou biometrické systémy, které mají za cíl autorizovat uživatele podle jeho hlasu. Skládají se ze dvou fází. První je zadání uživatele do databáze, tedy zaznamenání hlasu a jeho parametrů a jejich uložení. Druhou fází pak je pak autorizace uživatele při přihlášení do systému. Autorizace se skládá ze dvou kroků: identifikace řečníka (speaker identification) a ověření řečníka (speaker verification). Identifikace spočívá v porovnání hlasu osoby snažící se přihlásit do systému a vybrání uloženého uživatele s nejpodobnějšími charakteristikami hlasu. Ověření standardně probíhá určením míry shody mezi hlasem osoby snažící se přihlásit a vybraného uživatele z databáze např. pavděpodobnosti, že se opravdu jedná o daného člověka [26].



Obrázek 2.1: Principiální schéma ASV

První podstatnou záležitostí pro fungování ASV systémů je volba příznaků, tj. parametrů popisujících řeč daného člověka. Tyto příznaky by měly zahrnovat fyzikální vlastnosti hlasového traktu i naučené řečové charakteristiky. Dalším požadavkem je textová nezávislost. Mezi rozšířené příznaky patří: melovské kepstrální koeficienty (MFCC) [27], lineární kepstrální koeficienty (LFCC) [28], v posledních letech také: konstantní Q kepstrální koeficienty (CQCC)[29], gammatónové kepstrální koeficienty (GTCC) [30] či jiné nekepstrální příznaky.

Dalším důležitým bodem je volba modelu řečníka (speech embeding). Model řečníka se vytváří na základě extrahovaných příznaků při registraci uživatele do systému. Úkolem této části je převod příznaků, které mají různou velikost kvůli délce vstupní promluvy, na reprezentaci pevné délky, popisující řečníka. Promluvy stejného řečníka jsou si blízké. Dříve býval nejrozšířenější model založen na Gausovských směsí (GMM - z anglického *Gaussian Mixture Model*) [31],[32]. Později se začala využívat metoda podpůrných vektorů (SVM - z anglického *Support Vector Machines*) [33] a GMM-UBM (i-vectory). V dnešní době jsou rozšířené systémy založené na neuronových sítích např. na DNN (x-vektory) [34].

Posledním krokem je pak samotné ověření řečníka tzv. skórování. V této části jsou spolu porovnávány modely řečníků z databáze a model testovacího řečníka. Výstupem je skóre (pravděpodobnost). Na základě prahu pro skóre je následně rozhodnuto, zda se jedná o daného řečníka z databáze či nikoliv. Skóre může být vypočteno např. jako rozdíl logaritmických pravděpodobností, kosinovou vzdáleností (*cosine simiarity*), či pomocí pravděpodobnostní lineární diskriminativní analýzy (PLDA - z anglického *Probabilistic Linear Discriminant Analysis*) [35]. Tomuto kroku také mnohdy předchází redukce dimensionality za pomocí lineární diskriminativní analýzy (LDA - z anglického *Linear Discriminant Analysis*).

Podobně jako jiné biometrické systémy, jsou i ASV náchylné na útoky. Útoky lze v základu rozdělit na 3 kategorie a to Black box, White box a Gray box. V případě black box útoku nemá útočník žádné apriorní informace o fungování konkrétního ASV systému. Do této kategorie řadíme hardwarové útoky, **útoky s podvrženým hlasem**, a útoky na stabilitu systému. Jedná se o přímé útoky. Opakem je white box, kdy útočník má kompletní znalost a přístup do systému. Tento typ útoku je však málo pravděpodobný, pravděpodobnější je gray box tj. částečná znalost a přístup do systému a subsystémů. Mezi gray box útoky patří i útok s předvybráním oběti, kdy útočník s částečným přístupem do systému extrahuje informace o řečnících a s využitím podobné struktury ASV, jakou je napadená, vybere toho, na kterého se mu bude nejlépe útočit pomocí podvrženého hlasu. Útoků s různou mírou znalosti a přístupu do systému existuje mnoho, pro tuto práci jsou mnohé z nich irelevantní a jsou zde uvedeny jen pro ilustraci problematiky a nebudou zde dále rozváděny [36].

Blackbox útoky s podvrženým hlasem rozlišujeme základně tři. Prvním je prostá imitace člověkem. Náchylnost ASV systémů k tomuto útoku je však nejistá. Dalším a nejjednodušším způsobem je nahrání hlasu cílového řečníka a jeho následné přehrání ASV systému. Posledním je využití nástrojů TTS a VC.

Dále se útoky dělí na útoky virtuálního přístupu (LA - z anglického *Logical Access* a fyzického přístupu (PA - z anglického *Physical Access*). V LA scénaři je počítáno s tím, že útočník nemusí pro zadání hlasu do ASV systému využít mikrofon, postačí mu soubor s nahrávkou hlasu (reálného/umělého). Toto bývá obvykle v případech že ASV systém je vzdálený a ke komunikaci s ním je využito digitálního přenosového kanálu (telefon, VOIP). V PA scénáři musí útočník podvrženou nahrávku přehrát do mikrofonu ASV systému, či do mikrofonu, který je na vstupu digitálního přenosového kanálu, jestliže se do něj není schopen dostat přímo s digitální nahrávkou. V tomto případě je třeba v brát v potaz charakteristiky mikrofonu, reproduktoru, ze kterého je podvrh přehráván a prostředí. Tyto útoky jsou navíc špatně detekovatelné, k jejich provedení není třeba technických znalostí a jsou jednoduché na realizaci [37].

Samotné ASV systémy jsou schopné část útoků odfiltrovat a rovnou dané pokusy zamítnout. Toho je dosaženo již ve fázi trénování ASV systému a přidávání uživatelů. Systému jsou poskytnuty různé falešné nahrávky, z různých typů generování/úpravy hlasu i od různých mluvčích a jsou v systému označeny jako falešné. Z toho plyne, že pokud bude minimální vzdálenost modelu testovacího mluvčího a modelu jedné z podvržených ukázek, na jaké byl model trénován, dojde k zamítnutí. Toto řešení ovšem není optimální, jelikož dokáže pouze pro některé typy hlasů detekovat útok a to pouze pro konkrétní typ útoku, na které byl systém trénován, což je v době velkého rozšíření různých přístupů a metod strojově generované řeči nedostačující a neschopné dostatečné generalizace.

2.5 Detekce podvržené řeči

Detekce podvržené řeči je poměrně náročný úkol, jelikož detekční systémy musí být schopny pracovat s velikou škálou mluvčích a mnoha druhy útoků, které jsou obvykle pro systém neznámé tj. systém musí umět dobře generalizovat. Systémy pracující pouze s manuálním nastavováním mezí pro příznaky se ukázaly již jako nedostačující. V dnešní době se pro detekci využívají výhradně systémy strojového učení.

2.5.1 Jednodušší systémy

Jednodušší systémy jsou založeny na gaussovských směsích (GMM), kdy jsou příznaky (LFCC, MFCC ...) použity pro naučení dvou modelů. Jeden z modelů je naučen z příznaků přirozené řeči, druhý pak z příznaků podvržené řeči. Příznaky testovaného signálu jsou následně prezentovány oběma modelům, jejichž výstupem bývá log. pravděpodobnost. Tyto pravděpodobnosti jsou porovnány a model který dal vyšší, je označen za model odpovídající testovanému signálu [38].

Z jednoduchých klasifikačních algoritmů se také v malé míře objevují i SVM, ale v úpravě One Class SVM. Zde je při učení klasifikační hranice kladen důraz na naučení se podle jedné kategorie (přirozená řeč). Cokoliv pak nezapadá do této kategorie je označeno za kategorii druhou. Toto představuje rozdíl oproti tradičním SVM, která hranici budují rovnoměrně pro obě kategorie. [39].

2.5.2 Systémy založené na CNN

Další kategorii systémů představují založené na CNN. Jejich cílem je nalezení prostorových rozdílů mezi reálnými a podvrženými promluvami. U příznaků byly časové změny převedeny na prostorové. Sílou konvolučních vrstev je nalezení reprezentativních příznaků ze vstupních dat, které umožní klasifikaci. Vstupy ResNet bývají nejčastěji spektrogramy, či kepstrogramy (LFCC, MFCC), či Q koeficienty. Velmi často se také používají příznaky získané pomocí Wav2Vec 2.0. [40]. Nejpoužívanější z této kategorie jsou struktury ResNet. Tyto potenciálně hluboké neuronové sítě těží z přemostění mezi jednotlivými bloky, což vede ke snížení rizika vymizení, či exploze gradientu a síť tak může být hlubší. Struktury ResNet mohou nabývat nejrůznějších tvarů. Může se jednat o variace na sítě původně vytvořené pro klasifikaci obrázků tj. ResNet 18, ResNet 34, ResNet 50 a další [41] [42]. Také se může jednat o menší a jednodušší struktury obsahující pouze několik residuálních bloků [43].

2.5.3 Systémy zachycující časové změny

Další z kategorií systému nepřevádí časové změny na prostorové, ale využívá svých vnitřních pamětí pro zachycení časových změn. Mezi tyto sítě patří především systémy založené na RNN a především na LSTM. LSTM dokáží oproti RNN postihnout delší časový kontext a nejsou tak náchylné na problém s mizícím gradientem. Příklady využití RNN a LSTM zde: [44], [45].

2.5.4 Kombinované systémy

Další z možných přístupů je kombinace několika kategorií do jednoho systému. Jednou z častých kombinací je LCNN + LSTM tj. kombinace lehké konvoluční neuronové sítě a LSTM. Tato struktura je může být výhodná díky využití vlastností obou systémů. LCNN slouží pro extrakci NN příznaků z příznaků (např. spektrogramů) a LSTM detekuje časové změny v těchto vyextrahovaných příznacích [46].

Kromě uvedených systémů byly prováděny experimenty i dalších strukturách i kombinacích struktur. Většina ze systémů využívá na vstupu ručně počítané příznaky tj. spektra, kepstra a jiné, některé systémy ovšem pracují přímo s audiosignálem tzv. Rawwave systémy, které fungují end to end a extrakce jakýchsi příznaků je prováděna až strukturou sítě [47].

3 Dostupné datasety

Existuje několik datasetů sloužících pro trénování systémů pro detekci uměle syntetizované řeči, ať již pro ASV systémy, či pro jiná využití. V průběhu let databází přibývá a obsahují stále více nástrojů TTS a VC, což umožňuje detektorům lépe generalizovat. Problém stále je nedostatečná jazyková bohatost, jak je vidět v tabulce 3.1. Většina z datasetů je v angličtině či čínštině a využití systémů natrénovaných na těchto jazycích nemusí být optimální pro jazyky jiné.

Název	Jazyk	Generujících systémů	Počet promluv
ASVspoof15 [48]	Angličtina	10	263,151
ASVspoof19 LA [37]	Angličtina	19	121,461
ASVspoof21 LA [49]	Angličtina	13	164,612
ASVspoof21 DF [49]	Angličtina	100+	593,253
FakeAVCeleb [50]	Angličtina	1	11,857
FoR [51]	Angličtina	7	195,541
Voc.v [52]	Angličtina	8	82,048
In-The-Wild [53]	Angličtina	?	31,779
PartialSpoof [54]	Angličtina	19	121,461
WaveFake [55]	Angličtina, Japonština	9	136,085
ADD2022 [56]	Čínština	?	493,123
ADD2023 [57]	Čínština	?	517,068
FMFCC-A [58]	Čínština	13	50,000
HAD [59]	Čínština	2	160,836
CFAD [60]	Čínština	12	347,400
MLAAD [61]	38	82	154,000

Tabulka 3.1: Dostupné datasety. Přejato a přeloženo z [61]

Níže následují detailnější popisy tří z Datasetů, jednoho anglického: ASV Spoof 2019, který byl vybrán pro tuto práci, jednoho čínského: ADD 2023 a jednoho: vícejazyčného MLAAD. Nejdetailněji je zde rozepsán ASV Spoof 2019 dataset.

3.1 ASV Spoof dataset

První z této série datasetů vznikl v roce 2015 při příležitosti první ASV Spoof Challenge, která si klade za cíl vývoj co nejlepší metody detekce podvržené řeči pro ASV systémy [62]. Nejnovější kompletní dataset je ASV Spoof Dataset 2019, který obsahuje data pro trénování, úpravu i testování navrženého systému. Posledním datasetem je pak ASV Spoof 2021 dataset, který v LA scénáři primárně slouží pro testování modelů natrénovaných na předchozím datasetu (2019). Tento dataset přišel

také s Deepfake scénářem, kde nahrávky vznikají podobně jako u LA scénáře, je zde však více a on-date systémů.

ASV Spoof Dataset je složen ze dvou hlavních částí: PA scénáře a LA scénáře. PA se soustředí na útoky přehráním, kdy útočník má k dispozici nahrávku cíle svého útoku a přehraje je jí na vstup (mikrofon) ASV systému. Tato část datasetu pracuje s různými reprodukčními zařízeními a různými simulovanými RIR. Pro tuto práci není PA scénář důležitý.

LA obsahuje útoky strojově generovanou, či konvertovanou řečí. Dataset zahrnuje přirozené promluvy i promluvy generované různými nástroji TTS a VC.

3.1.1 LA scénář

Tato část obsahuje celkem 121 461 promluv (vět), které jsou jednotlivě uloženy jako ***.flac*** soubory do tří složek: trénovací, vývojové a hodnotící (testovací). Uměle vytvořené promluvy jsou utvořeny z přirozených promluv z databáze pomocí 19 systémů. Šest z těchto systémů jsou považovány za známé a slouží pro trénování a vývoj, 13 ostatních jsou neznámé a slouží pro hodnocení výkonu rozpoznávacího systému a jeho schopnosti generalizace. Základní rozdělení systému znázorněno v tabulce 3.2. Přirozené promluvy byly získány od 107 řečníků (46 mužů, 61 žen).

Dataset neobsahuje pouze soubory s promluvami, ale i složky s pomocnými ***.txt*** soubory, které obsahují informace o jednotlivých souborech a lze je využít pro trénování i následné testování či statistiky.

CM protocols

Tři soubory pro trénování, vývoj a testování, které obsahují základní informace o promluvách. Každý ze souborů obsahuje 5 sloupců. První sloupec je kód řečníka, druhý kód promluvy (název souboru promluvy bez přípony), třetí je zde nevyužit (pouze -), čtvrtý popisuje systém, kterým promluva vznikla ("-"- přirozená řeč, A01-A19 typ útoku), pátý pak obsahuje základní informaci, zda je promluva přirozená (bonafide), či podvržená (spoof).

ASV protocols

Soubory v této složce jsou rozděleny podle několika klíčů. Soubory jsou uloženy jako ASVspoof2019.LA.asv.<1>.<2>.<3>.txt, kde

1. obsahuje buď **dev** nebo **eval** v závislosti na tom, zda soubor popisuje vývojové, či evaluační promluvy

	Systém	Typ	Popis
Známé	A01	TTS	Neural waveform model
	A02	TTS	Vocoder
	A03	TTS	Vocoder
	A04	TTS	Waveform concatenation
	A05	VC	Vocoder
	A06	VC	Spectral filtering
Nenámé	A07	TTS	Vocoder+GAN
	A08	TTS	Neural waveform
	A09	TTS	Vocoder
	A10	TTS	Neural waveform
	A11	TTS	Griffin lim
	A12	TTS	Neural waveform
	A13	TTS/VC	Waveform concatenation+Waveform filtering
	A14	TTS/VC	Vocoder
	A15C	TTS/VC	Neural waveform
	A16	TTS	Waveform concatenation
	A17	VC	Waveform filtering
	A18	VC	Vocoder
	A19	VC	Spectral filtering

Tabulka 3.2: Rozdělení systémů generujících podvrženou řeč, podrobnější popis systémů: [37]

2. je buď **male (m)** nebo **female (f)** či **gender independent (gi)**. Toto rozděluje řečníky (promluvy) na muže a ženy. V gender independent souborech jsou za sebe seřazeny napřed mužské, následně ženské promluvy.

3. obsahuje **trl** - popis promluv, nebo **trn** - rozdělení promluv dle řečníka. Trl obsahuje i informace pro ASV systém, zda daná promluva, pokud se jedná o přirozenou patří do seznamu registrovaných uživatelů ASV.

ASV Scores

Složka rozdělena na dva soubory pro vývojovou a evaluační část. Každý ze souborů obsahuje následující informace o jednotlivých promluvách:

1. přirozené (bonafide) a uměle vytvořené (A01 - A19)

2. cílové - target (předkládaný řečník odpovídá tomu z databáze ASV), podvodný - nontarget (mluvčí mimo ASV databázi) a strojově podvržený - spoof.

3.2 ADD 2023 Challenge Dataset

Tento dataset vznikl v rámci Automatic Deepfake Detection Challenge 2023. Tato výzva se skládá ze tří částí a tomu odpovídá i rozložení datasetů. První z úkolů je detekce podvržené řeči, druhým detekce částí promluvy, která je podvržená, třetí pak na identifikaci typu systému generujícího umělou promluvu. Pro tuto práci je relevantní první a třetí část datasetu. Jazykem promluv v dataset je mandarínština.

Každá z částí datasetu je rozdělena na čtyři archivy: trénovací, vývojové a 2 testovací. Struktura archivu je přehledná a jednoduchá. Obsahuje složku s promluvami a **.txt** soubor s labely pro každou promluvu. Pro úkol detekce tento soubor obsahuje dva sloupce: první je název souboru s promluvou, druhý pak označení, zda se jedná o přirozenou promluvu (real), nebo podvrženou (fake). U třetího úkolu je druhý sloupec nahrazen číslem označující systém, který promluvu vygeneroval (0 - přirozená, 1-7 uměle generovaná).

Detekční část

Pro trénování bylo vybráno 3012 promluv od šedesáti řečníků, pro vývojovou fázi pak 2037 promluv od dalších 60 řečníků z řečového korpusu AISHELL-3 [63], jako přirozené promluvy. Pomocí systémů z tabulky 3.3 bylo vytvořeno 24 072 resp. 26 027 promluv od dalších 360 a 360 řečníků.

Systém	Popis
HiFiGAN	GAN, audio z mel-spektrogramů
LPCNet	WaveRNN vocoder, LPC+RNN
Multiband MelGAN	vocoder
StyleMelGAN	vocoder
Parallel WaveGAN	GAN vocoder + WaveNet
World	tradiční vocoder

Tabulka 3.3: Rozdělení systémů generujících podvrženou řeč, podrobnější popis systémů: [64]

Pro testovací datasety bylo vybráno 166 819 promluv od 1070 řečníků jako přirozené promluvy. Promluvy pochází nejen z datasetu AISHELL-3, ale také z AISHELL-1,THCHS-30 a dalších (celkem 8). Testovací dataset obsahující celkem 66 752 promluv byl vytvořen za pomocí 22 systémů (HiFiGAN, LPC Net, World, Alibaba, StyleMelGAN aj.) Více o systémech a databázích možno nalézt na [64].

Rozpoznávací část

Pro trénování a vývoj bylo využito 150 a 36 řečníků opět z databáze AISHELL-3. Pomocí nástrojů v tabulce 3.4 bylo vytvořeno 19 200 resp. 7 200 promluv. Dalších 3200 a 1200 bylo ponecháno jako přirozené.

Systém	Popis
Aliyun	komerční nástroj
DataBaker	TTS/VS, založeno na transformerech, komerční
Aispeech	TTS/VC, komerční
HiFiGAN	GAN, audio z mel-spektrogramů
WaveNet	generování akustického signálu přímo
World	tradiční vocoder

Tabulka 3.4: Rozdělení systémů generujících podvrženou řeč, podrobnější popis systémů: [64]

Testovací dataset vznikl taktéž za pomoci nástrojů z tabulky 3.4. Kromě nich byl využit i pro rozpoznávací systém neznámý nástroj Baidu. Testovací promluvy byly vygenerovány ve třech typech: čisté, zašuměné, s kompresí. Celkem dataset obsahuje 68 983 promluv od 440 řečníků.

3.3 MLAAD Dataset

4 Implementace

Pro celou práci byl využit programovací jazyk Python. Práce byla rozdělena do základních dvou bloků: do přípravné fáze a do fáze neuronové sítě. K tomuto bylo přistoupeno kvůli redukci výpočetního času pro trénování NN.

V přípravné fázi dochází k extrakci požadovaných příznaků z promluv a jejich následnému hierarchickému uložení. Hierarchické uložení je důležité pro další práci s daty. Byly využity následující Python knihovny:

1. **Soundfile**: Knihovna sloužící pro práci se zvukovými soubory. V práci využita pro načítání souborů s promluvami.
2. **Pandas**: Nástroj pro datovou analýzu a další práce s daty. Vhodný především pro data, která je možná reprezentovat 2D tabulkou (SQL databáze, CSV soubory a jiné). V práci využit pro uskladnění informací o jednotlivých souborech (promluvách), tj. jméno souboru, typ promluvy (přirozená/ uměle generovaná). Pandas je oproti reprezentaci dat ve 2D listu jednodušší na správu (přidávání, odebírání, vyhledávání) a je taktéž rychlejší.
3. **Numpy**: Nástroj pro práci s číselnými vektory maticemi i více dimenzionálními objekty. Oproti reprezentaci v listu je rychlejší.
4. **Spafe**: Knihovna pro práci se signály. Umožňuje extrakci různých příznaků: LFCC, MFCC, GTCC a jiné, dále pak preprocessing signálů, či filtrace a banky filtrů. [1]
5. **Pickle**: Nástroj pro ukládání dat jako objektů. Uloží např. celé numpy array jako jeden objekt do jednoho souboru, bez nutnosti ukládat postupně jednotlivé prvky z objektu. V práci využit pro ukládání souborů s vyextrahovanými příznaky.
6. **Threading**: Knihovna pro vícevláknové výpočty.

Extrakce příznaků byla prováděna paralelně pro trénovací, ověřovací a evaluační dataset.

Pro trénování (testování atd.) neuronové sítě byl využit framework PyTorch a další knihovny: Pandas, Numpy, Pickle, Matplotlib. Z frameworku Pytorch byly využity především následující části:

[1] Verze knihovny ke stažení se může lišit od verze v dokumentaci a je nutné si ověřit z kódu jaké argumenty a v jakém pořadí mají být dodány na vstup funkcí.

1. **torch**: Základní balíček umožňující práci s vícedimenzionálními tensory a matematické operace s nimi.

2. **torch.utils.data Dataset**: Je jedním z primitivních typů, které umožňují organizovanou a jednoduchou práci s vlastními i veřejně dostupnými organizovanými datasety. Výstupem jsou obvykle samply a labely. Pro práci s vlastním datasetem je nutné napsat třídu popisující a ovládající dataset, která bude z torch.utils.data Dataset dědit.

3. **torch.utils.data Dataloader**: Třída umožňující jednoduché načítání a práci s daty. Jestliže jsou data uložená pomocí torch.utils.data Dataset, je možné je obalit Dataloaderem, který je iterovatelný ve smyčce a umožňuje i rozdělení dat na dávky (batch) i zamíchání pořadí jednotlivých samplů a labelů (shuffle).

4. **torch.nn**: Modul obsahující třídy a funkcionality pro vytvoření a trénování neuronových sítí. Skládá se ze čtyř hlavních částí: Základní bloky (nn.Linear, nn.Conv2d, nn.Conv1d, nn.RNN, nn.LSTM, nn.BatchNorm2d...), aktivační funkce (nn.Relu, nn.Linear, nn.Tanh ...), ztrátové funkce (nn.CrossEntropyLoss, nn.MSELoss ...), struktury modelu (Sekvenční struktura pomocí kontejneru nn.Sequential, či vlastní struktura pomocí dědičnosti z nn.Module).

5. ...

4.1 Základní implementace pro ASV spoof dataset

Tato sekce obsahuje ukázky základní implementace pro ASV Spoof Dataset. Jedná se o zkrácené a jednodušší struktury kódu. Celý kód je k dispozici na `https://github.com/smidjir5/Detekce-podvrzene-reci`.

4.1.1 Dataset

Třída navržená pro práci s ASV Spoof 2019 datasetem pro LA i PA scénář. Vlastní třída Dataset dědí z torch.utils.data Dataset. V konstruktoru dochází k nastavení cest ke vstupním příznakům (příznaky vyextrahované z **.flac** souborů a objektově uložené jako numpy array ve stejné hierarchii). Dále se nastavují cesty pro ovládací soubory (CM protocols), délka příznaků, scénář (LA,PA), využití prvního příznaku (u kepstra C0) a způsob doplnění příznakového tensoru (2D matice), jestliže by byla kratší než je nastavená délka příznaků.

Dále jsou definovány tagy pro jednotlivé scénáře. Pro LA scénář jsou tagy brány jako systém, který vygeneroval promluvu, či přirozená řeč. Tyto stringy ("–", "A01"..."A19")

je vhodné převést do číselné reprezentace. Dále jsou do číselné reprezentace převedeny i labely (přirozené - bonafide, podvržené - spoof).

```
if self.access_type == 'LA':
    self.tag = {"-": 0, "A01": 1, "A02": 2, "A03": 3,
    "A04": 4, "A05": 5, "A06": 6, "A07": 7, "A08": 8,
    "A09": 9, "A10": 10, "A11": 11, "A12": 12, "A13": 13,
    "A14": 14, "A15": 15, "A16": 16, "A17": 17, "A18": 18,
    "A19": 19}
else:
    self.tag = {"-": 0, "AA": 1, "AB": 2, "AC": 3, "BA": 4,
    "BB": 5, "BC": 6, "CA": 7, "CB": 8, "CC": 9}
self.label = {"spoof": 1, "bonafide": 0}
```

Následně dochází k načtení protokolového (ovládacího) souboru pomocí `pandas`. Ten je reprezentován jako `pandas dataframe` o čtyřech sloupcích.

Třída následně obsahuje dvě metody, které každá třída dědící z `torch.utils.data Dataset` musí obsahovat a to `__len__(self)`, která vrací počet prvků v datasetu, a `__getitem__(self,idx)`, která vrací i-tý sample a label. Z pandas dataframu se vybere i-tý řádek, z něj se vezme sloupec se jménem, to se spojí s příponou, vytvoří se cesta k souboru a soubor je pomocí `pickle.load` načten. Label se opět vezme z dataframu.

4.1.2 Model

Model neuronové sítě je taktéž reprezentován třídou, která dědí z `nn.Module`. V konstruktoru dochází k inicializaci jednotlivých komponent sítě a povinné metodě `forward` pak k jejich seřazení a napojení na sebe.

```
class Basic_Block(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv_layer = nn.Conv2d(in_channels=32,out_channels=32,
        kernel_size=(3,3),padding=1,stride=1)
        self.conv_layer2 = nn.Conv2d(in_channels=32,out_channels=32,
        kernel_size=(3,3),padding=1,stride=1)
        self.bn_layer = nn.BatchNorm2d(32)
        self.relu = nn.LeakyReLU()

    def forward(self,x):
        out = self.conv_layer(x)
```

```
        out = self.bn_layer(out)
        out = self.relu(out)
        out = self.conv_layer2(out)
        out = out + self.conv_layer2(x)
        out = self.bn_layer(out)
        out = self.relu(out)
        return out
```

Objekty vytvořené z takto definovaných tříd pak lze použít jako komponenty do dalších struktur.

4.1.3 Trénování

Trénovací soubor je navržen tak aby ho nebylo nutné spouštět z IDE, ale i z příkazové řádky. Všechna potřebná nastavení lze udělat pomocí přepínačů. K tomuto a jednoduché práci s argumenty byl využit modul `argparse`, konkrétně `argparse.ArgumentParser()`, viz ukázka níže. Výstupem je pak objekt k jehož proměnným (lr,bs,epochs ...) se lze dostat pomocí tečkové notace. Tento objekt je používán jako nostná struktura i pro další funkce v souboru, které z něj čtou, nebo do něj přidávají.

```
parser = argparse.ArgumentParser()
parser.add_argument("--lr", type=float,default=5e-5)
parser.add_argument("--bs",type=int,default=32)
parser.add_argument("--epochs",type=int,default=25)
args = parser.parse_args()
```

Dále probíhá inicializace. V této fázi se ověřuje zda je možné na daném zařízení využít trénování na GPU (CUDA). Jestliže ano, je nastavena proměnná `args.device` na cuda. Při práci s neuronovou sítí i s jednotlivými tenzory či datasety je nutné u nich pro správné fungování i na CPU i na GPU nastavit parametr `device`.

```
args.cuda = torch.cuda.is_available()
print("Cuda device available: ", args.cuda)
args.device = torch.device("cuda" if args.cuda else "cpu")
```

Poté jsou nastaveny cesty k protokolovým souborům, základní cesty k databázi a je vytvořena složka pro uložení modelu a následných pomocných, informativních a popisných souborů.

Následuje inicializace Datasetů (trénovacího a vývojového), Dataloaderů a modelu neuronové sítě.

```
training_set = ASV_Spoof_2019_dataset(...)
validation_set = ASV_Spoof_2019_dataset(...)

train_data_loader = DataLoader(training_set,args.bs,
shuffle=True)
validation_data_loader = DataLoader(validation_set,args.bs,
shuffle=True)

model = Residual_Net(...).to(args.device)
```

Dále je nastaven optimalizátor a ztrátová funkce. Poté již začíná trénovací smyčka. Hlavní smyčka prochází přes epochy, první vnořený cyklus pak přes jednotlivé dávky (batch) trénovacího dataloaderu. Zde je ovšem nutné upravit dimenze samplů z Dataloaderu před vstupem do NN. Uvažujme uložený a následně načtený, zkrácený, či prodloužený kepstrogram o velikosti 19 x 200 a velikost dávky 32. Výstupem z dataloaderu je pak tensor o velikosti 32 x 19 x 200. Konvoluční vrstvy však požadují vstup ve formátu: Velikost dávky (batch size) x počet kanálů x výška x šířka. Toto je způsobeno tím, že konvoluční vrstvy pracují s více konvolučními jádry a výsledek konvoluce s každým jádrem má svůj kanál. Je třeba tedy přidat jednu prázdnou dimenzi mezi 32 a 19, aby formát byl 32 x 1 x 19 x 200 (metoda `unsqueeze`) tj. jeden kanál na vstupu.

```
lfcc = lfcc.unsqueeze(1).float().to(args.device)
```

Jednotlivé predikce modelu pro každý z batchů jsou ukládány do proměnných a jsou počítány i statistiky (ACC, loss a další viz Kapitola 5). Vše je následně uloženo. Toto se děje i pro validační dataset, ovšem model je v tuto chvíli nastaven na evaluační fázi a nedochází zde ke změnám v gradientu.

4.2 Implementace pro Metacentrum

Metacentrum [65] je gridová distribuovaná výpočetní infrastruktura umožňující provádět velmi náročné výpočty, které by nebylo možné z důvodu nedostatečné kapacity, výkonu či strojového času počítat na desktopech nebo počítačových strukturách vědeckých týmů. Umožňuje zadávat interaktivní i dávkové úlohy. Vše je ovládáno z příkazové řádky a pomocí bash scriptů přes SSH spojení. Metacentrum nabízí také rozměrné úložiště pro skripty i datasety. Nahrání a manipulace se soubory jsou možná pomocí FTP. Výpočty jsou podporované jak na CPU, tak i na GPU.

Metacentrum podporuje výpočty v Pythonu a veliké množství knihoven je obsaženo v jednotlivých modulech [66]. Pro standardní sériové CPU výpočty postačí pouze přidat do .sh skriptu, který ovládá a spouští Python skript, daný modul pomocí příkazu `module add xxx`. Jestliže požadovaná knihovna není v žádném z modulů, je možné jí ve scriptu nainstalovat pomocí `pip install`.

Starší verze frameworku Pytorch jsou taktéž dostupné jako moduly. Jejich použití však není doporučováno. Doporučené je využití tzv. NVIDIA kontejnerů, což je přenosné software prostředí optimalizované pro běh na GPU. Kontejnery v sobě obsahují vše potřebné pro běh Pytorch, i další knihovny. Tyto kontejnery se pro dávkové soubory standardně spouští pomocí:

```
singularity run -nv /cvmfs/singularity.metacentrum.cz/...
.../NGC/PyTorch:24.10-py3.SIF my_python_script.py,
```

kde cvmfs/singularity.metacentrum.cz//NGC/PyTorch:24.10-py3.SIF je název a umístění kontejneru.

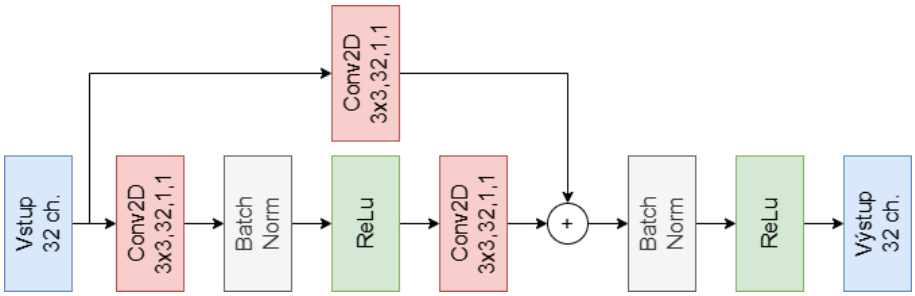
Jestliže je v kódu třeba využít knihovnu, která není obsažena v kontejneru, je nutné kód upravit. Jelikož se jedná o svébytné prostředí, využití `pip install` mimo kontejner nebude fungovat. Existují dvě možnosti řešení. První a složitější je zkopírování kontejneru, jeho otevření (vytvoření sandboxu), doplnění dalších knihoven a jeho opětovné zavření. Toto je na Metacentru poměrně komplikovaný proces s mnoha problémy. Řešení v podobě vytvoření, či replikace kontejneru staženého přímo od NVIDIA, také nelze začínajícím uživatelům doporučit. Druhá možnost je využití `singularity exec` místo `singularity run`. Exec narozdíl on run, které spustí pouze defaultní akci kontejneru (v tomto případě python script), umí spustit příkazy uvnitř kontejneru, tj. je možné využít pip uvnitř kontejneru a pomocí něj nainstalovat příslušné knihovny. Zde je nutné brát v potaz, že pip se nevyskytuje ve všech kontejnerech a na všech strojích na kterých je výpočet prováděn. Proto je nutné pip nainstalovat (např. pomocí souboru get-pip.py). Dále je nutné zohlednit, že snaha instalovat pip do míst, kde již instalován je, povede k pádu programu. Řešení může vypadat následovně:

```
singularity exec -nv /cvmfs/singularity.metacentrum.cz/NGC/PyTorch:
24.10-py3.SIF  bash -c "if ! command -v pip &> /dev/null; then
python get-pip.py; fi && pip install torchsummary && pip install argparse
&& python my_script.py"
```

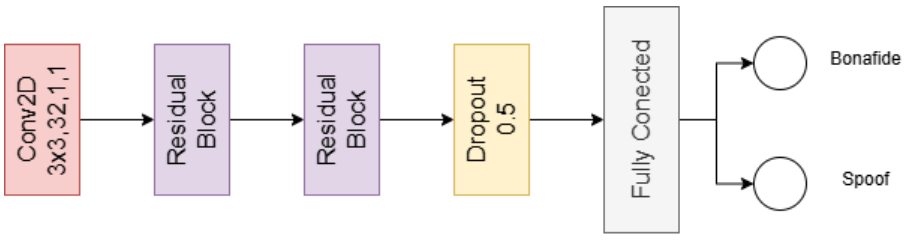
5 Experimenty a výsledky

První experimenty měly za cíl vybudovat infrastrukturu celého projektu tj. kód pro extrakci a uložení příznaků, třídy pro práci s datasety, kód pro trénování, evaluaci a testování, kód pro výpočet statistik atd. Všechny kódy byly koncipovány od začátku tak, aby nebylo jejich spuštění možné pouze z IDE s nutností přepisovat proměnné, ale aby je bylo možné spustit z příkazové řádky, či bash scriptem a to nejen na PC, ale i na gridových výpočetních službách jako je Metacentrum, kam byl kód postupně přesouván. Dalším cílem bylo seznámení se postupy implementace, ovládání a korigování pokročilejších struktur NN než jsou DNN pouze s dopřednými vrstvami.

Pro první experimenty byl využit ASV Spoof dataset. Jako příznaky byly zvoleny LFCC, konkrétně 19 kepstrálních koeficientů (c_1-c_{19}), bez delta koeficientů z frekvenčního rozsahu 0 - 4kHz. Jako struktura sítě byl zvolen dvou-blokový ResNet s dropouty nastavenými na 30 % a s 800 neurony v plně propojené vrstvě. Struktura sítě znázorněna na obrázku 5.1. LR nastaven na 1e-4 s ad-hoc schodovým klesáním. Prvky sítě byly inicializovány náhodně.



(a) Schéma základního reziduálního bloku. Konvoluční vrstvy: Jádro 3x3, 32 kanálů, Stride 1, Padding 1. Konvoluce v přemostění není v tomto nastavení nutná, později poslouží pro srovnání dimenzí. Možno nahradit převzorkováním.



(b) Schéma základní sítě

Obrázek 5.1: Základní struktura použité sítě

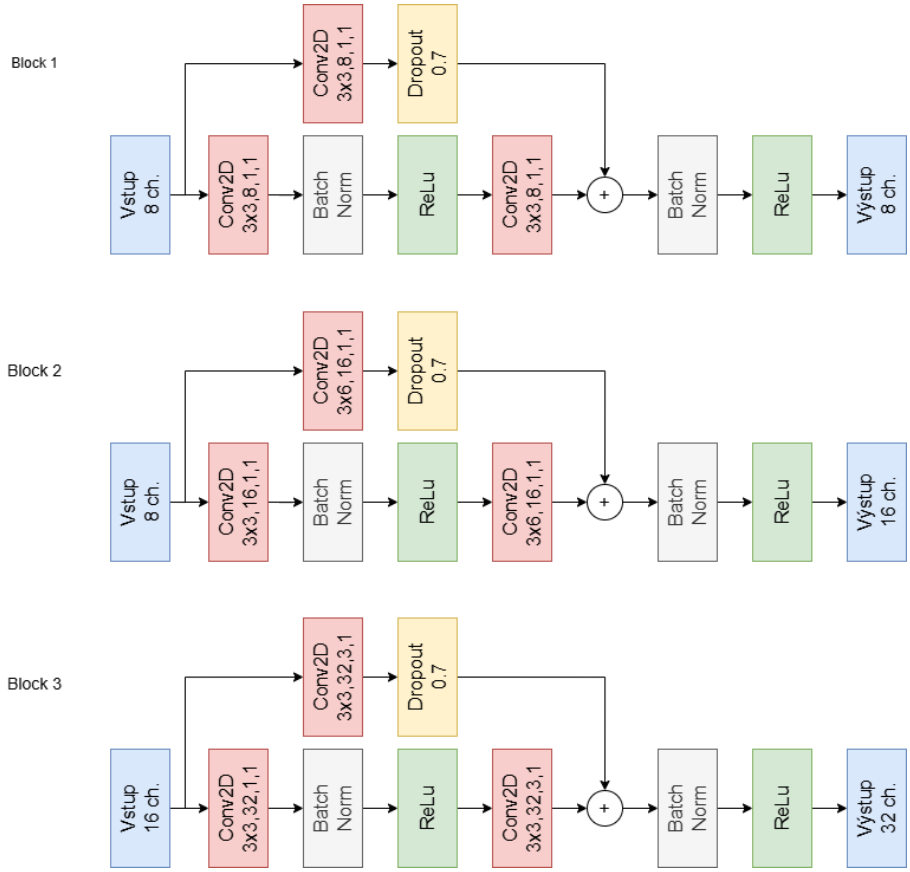
Jako metrika úspěšnosti učení byla zprvu vybrána ACC. Z pohledu této metriky síť již po několika epochách vykazovala velmi dobré výsledky (ACC kolem 90 %) pro všechny tři datasety. Následnou analýzou výstupů byla zjištěna příčina, která spočívala v tendenci sítě označit všechny vstupy za podvržené. Jak již bylo zmíněno podvržených promluv je v datasetu přibližně 9x více, tudíž 90% přesnost predikce (ACC) byla zapříčiněna tímto.

Aby bylo možné toto sledovat a následně korigovat, byly zavedeny další metriky a to: FPR (False Positive Rate) a FNR (False Negative Rate). Následně se ještě počítá EER (Equal Error Rate) a F1 skóre. Jestliže je FPR vysoké, znamená to, že systém klasifikuje příliš mnoho přirozených promluv jako podvržené. Optimální poměr mezi FPR a FNR záleží na konkrétním využití, a může být předmětem diskuze. V této části práce bylo cílem obě dvě hodnoty zachovat vyrovnané.

Jedním z kroků, mírnící nevyváženost datasetu je přidání vah ke ztrátové funkci Corss Entropy. Tyto váhy určují, jak moc má být brána v potaz každá z klasifikačních kategorií. Optimální hodnota se ukázala být 3-4:1 ve prospěch přirozených promluv. Takto optimalizovaná síť dosáhla EER = 30%.

Další pokusy měly za cíl postupné zlepšování výsledků sítě. K LFCC (bez c_0) byly přidány Δ a $\Delta\Delta$ koeficienty. Struktura sítě byla zvětšena na 4 bloky a bylo experimentováno i se 64 kanály u některých konvolučních vrstev. Toto vedlo k delšímu výpočetnímu času a ke zlepšení výsledků na trénovací množině. Na vývojové a testovací se opět výrazně zvýšila FP. Síť byla tudíž velmi přetrénovaná na daná data a nedokázala dobře generalizovat (overfitting).

Jako řešení se ukázalo zmenšení počtů kanálů a jejich postupný růst a využití i asymetrických konvolučních jader (Delší v časové dimenzi). Dalším podstatným zjištěním bylo, že síť velmi využívá při své práci přemostění, jehož účelem je pouze zamezení ztrátě velikosti gradientu při zpětné propagaci, i pro klasifikaci. Z tohoto důvodu byl k přemostění přidán dropout ve výši 70 %. Toto mělo zásadní vliv. Dále došlo k využití větších posunů (stride) některých konvolučních jader. Jako optimalizační algoritmus byl využit Adam s využitím L2 normalizace poklesu vah (weight decay) = 1e-9. Struktura bloků této sítě znázorněna na obrázku 5.2. Další nastavení jsou popsána v tabulce 5.1 a výsledky na evaluačním datasetu v tabulce 5.2.



Obrázek 5.2: Znázornění tří bloků Resnetu

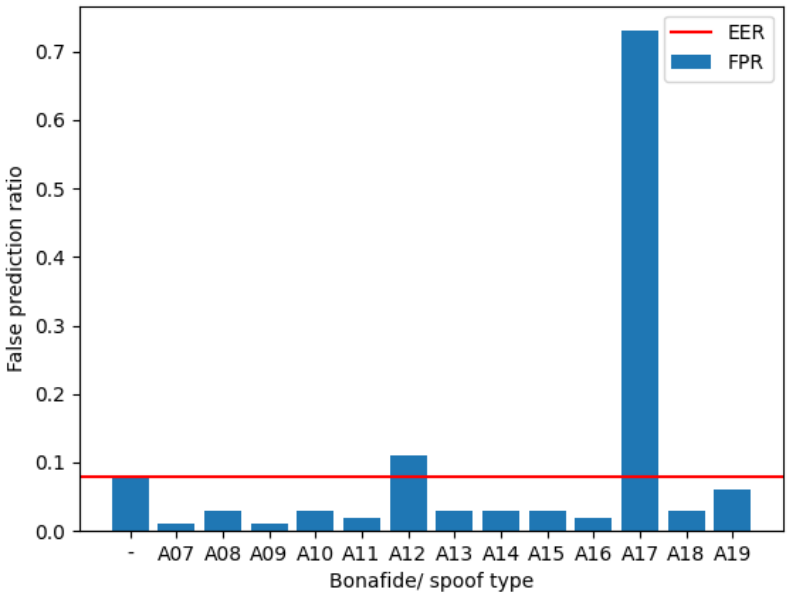
Parametr	Nastavení
Příznaky	LFCC
Dimenze příznaků	(3*18) x 200
Délka okna segmentace	
Preemfáze	Ano
Δ ΔΔ	Ano
LR	5e-5
Pokles LR	Schodovitý
Optimalizační alg.	Adam
Velikost dávky	32
Epochy	25
Váhy pro Ztr. funkci (bonafide:spoof)	3:1
Neuronů v lin. vrstvě	800

Tabulka 5.1: Základní parametry sítě

Výsledky ukazují 8% chybovost a vyváženou klasifikaci pro obě dvě skupiny. Z tohoto je patrná poměrně dobrá schopnost generalizace. Výsledky pro jednotlivé podskupiny jsou zobrazeny na obrázku 5.3. Z průměru se velmi odchyluje systém 17.

TP	TN	FP	FN	EER	F1
0.91	0.92	0.08	0.09	0.08	0.92

Tabulka 5.2: Výsledky upravené sítě



Obrázek 5.3: Výsledky pro přirozené promluvy a jednotlivé systémy TTS a VC

6 Následující práce

Jak již bylo zmíněno v Úvodu, cílem navazující práce je tvorba nástroje pro detekci podvržené řeči v českém jazyce. Pro tento úkol je třeba dataset s přirozenými i podvrženými promluvami v češtině. V době psaní práce není autorovi známa žádná takováto databáze, která by byla volně dostupná či dostupná pro akademickou obec. Z toho důvodu je třeba takovýto dataset zhotovit. Nutná velikost datasetu se ukáže až v průběhu navazující práce. V prvním kroku je třeba sestavit dataset, který bude sloužit jako testovací pro již natrénované modely na ASV spoof datasetu. Výsledky pak rozhodnou, zda bude nutné databázi rozšířit i o trénovací a validační část. Toto bude záviset na schopnosti modelu generalizovat pro jiný jazyk a jiné nástroje, tvořící promluvy.

6.1 Databáze promluv v českém jazyce

Existuje několik databází promluv v českém jazyce. Níže jsou uvedené.

SpeeCon Dataset obsahující nahrávky od 640 mluvčích. Nahrávky jsou ve formě krátkých promluv (vět) a existuje k nim i přepis [67].

Czech SpeechDat(E) Database Obsahuje velmi krátké promluvy (slova) od 1052 řečníků. K promluvám je taktéž k dispozici přepis [68].

Common Voice dataset Pravděpodobně největší mnohojazyčná databáze na světě. Její česká část obsahuje 266 hodin od 1 035 řečníků [69].

Voxpopuli Dataset obsahující nahrávky s přepisem i bez přepisu pořízený ze záznamů z Europarlamentu mezi lety 2009 až 2020. Obsahuje také v českém jazyce.

6.2 Nástroje pro tvorbu datasetu

Následující nástroje mohou být využity pro vytvoření českého datsetu s podvrženými promluvami. Jsou zde uvedeny pouze nástroje, které umožňují vložení požadovaného řečníka (speaker embeding) při generování promluv. Nástroje umožňující využít pouze pevně danou množinu naučených řečníků zde uvedeny nejsou, byť je zde šance na jejich využití, při potenciální nedostatku dalších nástrojů. Níže následuje výčet předtrénovaných nástrojů pro český jazyk.

Předtrénovaný Speech T5 U tohoto modelu bylo natrénováno pouze jeho jádro a je třeba ho dotrénovat pro TTS. [6]. Přístup možný přes `transformers` od Hugging Face.

Dotrénovaný (Fine-tuned) Speech T5 Tento TTS [7] využívá standardní anglické jádro Speech T5, ale byl dotrénovaný na české části Vox Populi [8] datasetu. Přístup možný přes `transformers` od Hugging Face [70].

XTTS Multilinguální model podporující i češtinu od Cocqui AI [71]. Přístup možný přes `transformers` od Hugging Face, nebo přes knihovnu `TTS` od Cocqui.

Všechny výše uvedené nástroje lze ještě dotrénovat na dalších datsetech. Další možností je využití systémů, které pro češtinu trénované nebyly, ale lze je přetrénovat, či dotrénovat. Zde připadá v úvahu využití Speech T5, Tacontron 2, či dalších (viz Sekce 2.2.2 a jejich dotrénování na českých datasetech. Všechny tyto modely jsou přístupné přes knihovnu (API) `transformers`, čož usnadňuje implementaci. Jak již bylo zmíněno, tyto modely jsou schopné práce s obyčejným textem na vstupu, ale toto může být nevýhodné u číslic a specifických znaků pro češtinu (ě, š, č, ...). Z tohoto důvodu přichází v potaz využití systému eSpeakNG [72], který v sobě zahrnuje i konvertor do fonetického přepisu a podporuje i češtinu, či `TTS.tts.utils.text.phonemizers` od Cocqui TTS, který z eSpeaku vychází. Dotrénování u některých dalších modelů zařazených do Cocqui TTS (VITS, Glow TTS) taktéž přichází v úvahu. Další z možností zvětšení datasetu je využití česky generovaných promluv z MLAAD datasetu [61].

6.3 Úkoly a harmonogram pro navazující práci

Jedním z prvních úkolů je stažení zahraničních datasetů popsaných v tabulce 3.1. Dále bude nutné tyto datasety standardizovat, co se hierarchie a uložení labelů týče, či napsání několika tříd pro práci s těmito datasety.

Dalším z úkolů je sestavení českého datasetu podvržené řeči. To spočívá nejprve v získání českých databází přirozené řeči. Následně budou využity předtrénované nástroje pro vygenerování první části datasetu. Dobré bude také uvážit znění chtěných promluv, zda využít přepisy z již existujících či vytvořit i nové. Dalším krokem je pak natrénování, či dotrénování uvedených TTS modelů a jejich využití pro generování. Dalším potenciálním úkolem bude nalezení dalších nástrojů pro zajištění co možná největší pestrosti datasetu. Primárním obdobím práce na datasetu by měl být únor a začátek března.

V březnu a začátkem dubna je plánováno nalezení a implementace několika struktur pro rozpoznání podvržené řeči. Natrénované modely na různých zahraničních datasetech budou porovnány mezi sebou ve schopnosti generalizace a budou otestovány i na českém datasetu. Následovat by mělo natrénování modelu na českém datasetu.

Zbylý čas bude využit k porovnání výsledků, úpravě modelů a také k sepsání textu Diplomové práce.

7 Závěr

Tato práce se zabývala problematikou podvržené řeči. V rámci teoretické části byly popsány základní informace o strojově generované řeči, jejím využití i zneužití. Byly popsány dnes využívané systémy, jejichž využití připadá v potaz v navazující práci. Dále byla provedena rešerše dostupných datasetů zbývajících se detekcí podvržené řeči.

Praktická část měla za úkol seznámit autora s implementací sítí složitějších než DNN ve frameworku Pytorch a vybudování základní struktury umožňující pracovat s datasety, extrahovat příznaky, natrénovat a otestovat síť a následně vytvořit výstupy ve formě statistik. V rámci této části bylo experimentováno s jednoduchým tříblokovým ResNetem na ASV Spoof 2019 datasetu. Přes malou velikost sítě, byl model schopen poměrně přijatelných výsledků a byl schopen generalizace. Jako nástroj pro vybudování infrastruktury a naučení se ovládání a nastavování sítí posloužil sobře.

Poslední část se zabývala možností využití detekce podvržené řeči pro český jazyk. Rešerše ukázala dostupnost několika datasetů obsahujících české promluvy, které půjdou využít pro tvorbu české databáze podvržené řeči. Dále byly ukázány nástroje TTS a bylo popsáno, jak tyto nástroje využít pro český jazyk. Následně byl prezentován očekávaný harmonogram a postup následující práce.

Celkově práce splnila svůj účel a zadání a může sloužit jako odrazový bod pro navazující práci. Nalezené teoretické informace jsou zásadní pro pokračování, a zkušenosti nabyté v praktické části se budou také velmi hodit.

Bibliografie

[1] J. Flanagan, *Speech Analysis, Synthesis and Perception*. Springer-Verlag Berlin Heidelberg GmbH, 1972.

[2] K. e. Tokuda, „Speech Synthesis Based on Hidden Markov Models“, *Proceedings of the IEEE*, 2013.

[3] P. Pollák, „BE2M31ZRE - Speech Synthesis“, University Lecture at FEE CTU Prague, 2024.

[4] J. Ao, R. Wang, L. Zhou et al., *SpeechT5: Unified-Modal Encoder-Decoder Pre-Training for Spoken Language Processing*, 2022. arXiv: 2110.07205 [eess.AS]. URL: https://arxiv.org/abs/2110.07205.

[5] R. Prenger, R. Valle a B. Catanzaro, *WaveGlow: A Flow-based Generative Network for Speech Synthesis*, 2018. arXiv: 1811.00002 [cs.SD]. URL: https://arxiv.org/abs/1811.00002.

[6] „SpeechT5-base-cs-tts“. (), URL: https://huggingface.co/fav-kky/SpeechT5-base-cs-tts (cit. 11. 01. 2025).

[7] „Speecht5 finetuned voxpopuli cs“. (), URL: https://huggingface.co/kfahn/speecht5_finetuned_voxpopuli_cs/tree/main (cit. 11. 01. 2025).

[8] C. Wang, M. Rivière, A. Lee et al., *VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation*, 2021. arXiv: 2101.00390 [cs.CL]. URL: https://arxiv.org/abs/2101.00390.

[9] „Zero-Shot vs. Few-Shot Multi-Speaker TTS Using Pre-trained Czech SpeechT5 Model“. (), URL: https://jalehecka.github.io/TSD2024/ (cit. 11. 01. 2025).

[10] J. Lehečka, Z. Hanzlíček, J. Matoušek a D. Tihelka, „Zero-Shot vs. Few-Shot Multi-speaker TTS Using Pre-trained Czech SpeechT5 Model“, in *Text, Speech, and Dialogue*. Springer Nature Switzerland, 2024, s. 46–57, ISBN: 9783031705663. DOI: 10.1007/978-3-031-70566-3_5. URL: http://dx.doi.org/10.1007/978-3-031-70566-3_5.

[11] „Tacotron (/täkōträn/): An end-to-end speech synthesis system by Google“. (), URL: https://google.github.io/tacotron/index.html (cit. 11. 01. 2025).

[12] Y. Ren, Y. Ruan, X. Tan et al., *FastSpeech: Fast, Robust and Controllable Text to Speech*, 2019. arXiv: 1905.09263 [cs.CL]. URL: https://arxiv.org/abs/1905.09263.

[13] „How to use text-to-speech with low vision“. (), URL: https://www.perkins.org/resource/ways-to-read-webpages-without-a-traditional-screen-reader/ (cit. 16. 11. 2024).

[14] „The 2024 Guide To Chatbots In Banking“. (), URL: https://springsapps.com/knowledge/the-2024-guide-to-chatbots-in-banking (cit. 16. 11. 2024).

[15] S. O. Arik, J. Chen, K. Peng, W. Ping a Y. Zhou, *Neural Voice Cloning with a Few Samples*, 2018. arXiv: 1802.06006 [cs.CL]. URL: https://arxiv.org/abs/1802.06006.

[16] H.-T. Luong a J. Yamagishi, „NAUTILUS: A Versatile Voice Cloning System“, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, roč. 28, s. 2967–2981, 2020. DOI: 10.1109/TASLP.2020.3034994.

[17] „Unusual CEO Fraud via Deepfake Audio Steals US243, 000*FromUKCompany*“. (), URL: https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/unusual-ceo-fraud-via-deepfake-audio-steals-us-243-000-from-u-k-company (cit. 17. 11. 2024).

[18] „AI voice cloning is used in a huge heist being investigated by Dubai investigators, amidst warnings about cybercriminal use of the new technology.“ (), URL: https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=5f1a42037559/ (cit. 17. 11. 2024).

[19] „https://www.theguardian.com/technology/article/2024/may/10/ceo-wpp-deepfake-scam“. (), URL: https://www.theguardian.com/technology/article/2024/may/10/ceo-wpp-deepfake-scam (cit. 17. 11. 2024).

[20] „Údajná nahrávka telefonátu predsedu PS a novinárky Denníka N vykazuje podľa expertov početné známky manipulácie“. (), URL: https://fakty.afp.com/doc.afp.com.33WY9LF (cit. 16. 11. 2024).

[21] „A fake recording of a candidate saying he'd rigged the election went viral. Experts say it's only the beginning“. (), URL: https://edition.cnn.com/2024/02/01/politics/election-deepfake-threats-invs/index.html (cit. 16. 11. 2024).

[22] „Deepfake audio of Sir Keir Starmer released on first day of Labour conference“. (), URL: https://news.sky.com/story/labour-faces-political-attack-after-deepfake-audio-is-posted-of-sir-keir-starmer-12980181 (cit. 16. 11. 2024).

[23] „S falešnými videi se roztrhl pytel, zneužívají i značku CNN Prima NEWS. Jak se jim bránit?“ (), URL: https://cnn.iprima.cz/deepfake-cnn-prima-news-falesne-video-podvod-jak-je-poznat-439783 (cit. 12. 01. 2025).

[24] „ANALÝZA: Rozdám 50 milionů korun, slibuje ve falešném videu Procházka. Jak poznat podvod?“ (), URL: https://cnn.iprima.cz/analyza-jiri-prochazka-50-milionu-korun-podvod-umela-inteligence-437902 (cit. 12. 01. 2025).

[25] „MrBeast calls TikTok ad showing an AI version of him a 'scam'“. (), URL: https://www.nbcnews.com/tech/mrbeast-ai-tiktok-ad-deepfake-rcna118596 (cit. 12. 01. 2025).

[26] „Automatic speaker verification systems“. (), URL: https://antispoofing.org/automatic-speaker-verification-systems/#speaker-identification-vs-speaker-verification (cit. 17. 11. 2024).

[27] Z. K. Abdul a A. K. Al-Talabani, „Mel Frequency Cepstral Coefficient and its Applications: A Review“, *IEEE Access*, roč. 10, s. 122 136–122 158, 2022. DOI: 10.1109/ACCESS.2022.3223444.

[28] P. Sovka a P. Pollák, *Vybrané metody číslicového zpracování signálů*. Vydavatelství ČVUT, 2003, ISBN: 80-01-02821-6.

[29] B. Blankertz. „The Constant Q Transform“. (), URL: https://doc.ml.tu-berlin.de/bbci/material/publications/Bla_constQ.pdf (cit. 17. 11. 2024).

[30] X. Valero a F. Alías-Pujol, „Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification“, *Multimedia, IEEE Transactions on*, roč. 14, s. 1684–1689, pros. 2012. DOI: 10.1109/TMM.2012.2199972.

[31] S. S. Jagtap a D. Bhalke, „Speaker verification using Gaussian Mixture Model“, in *2015 International Conference on Pervasive Computing (ICPC)*, 2015, s. 1–5. DOI: 10.1109/PERVASIVE.2015.7087080.

[32] B. Blankertz. „The Constant Q Transform“. (), URL: https://doc.ml.tu-berlin.de/bbci/material/publications/Bla_constQ.pdf (cit. 17. 11. 2024).

[33] W. M. Campbell, J. P. Campbell, T. P. Gleason, D. A. Reynolds a W. Shen, „Speaker Verification Using Support Vector Machines and High-Level Features“, *IEEE Transactions on Audio, Speech, and Language Processing*, roč. 15, č. 7, s. 2085–2094, 2007. DOI: 10.1109/TASL.2007.902874.

[34] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey a S. Khudanpur, „X-Vectors: Robust DNN Embeddings for Speaker Recognition“, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. DOI: 10.1109/ICASSP.2018.8461375.

[35] S. Ioffe, „Probabilistic Linear Discriminant Analysis“, in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof a A. Pinz, ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, s. 531–542, ISBN: 978-3-540-33839-0.

[36] P. Gupta, H. A. Patil a R. C. Guido. „Vulnerability issues in Automatic Speaker Verification (ASV) systems“. (2014).

[37] X. Wang, J. Yamagishi, M. Todisco et al., *ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech*, 2020. arXiv: 1911.01601.

[38] A. S. consorcium. „LFCC-GMM baseline model“. (), URL: https://www.asvspoof.org/asvspoof2015/I3A.pdf (cit. 08. 01. 2025).

[39] J. Villalba, A. Miguel, A. Ortega a E. Lleida. „Spoofing Detection with DNN and One-class SVM for the ASVspoof 2015 Challenge“. (), URL: https://www.asvspoof.org/asvspoof2015/I3A.pdf (cit. 08. 01. 2025).

[40] A. Baevski, H. Zhou, A. Mohamed a M. Auli, *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020. arXiv: 2006.11477 [cs.CL]. URL: https://arxiv.org/abs/2006.11477.

[41] K. He, X. Zhang, S. Ren a J. Sun, *Deep Residual Learning for Image Recognition*, 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[42] Y. Zhang, F. Jiang a Z. Duan, „One-Class Learning Towards Synthetic Voice Spoofing Detection“, *IEEE Signal Processing Letters*, roč. 28, s. 937–941, 2021, ISSN: 1558-2361. DOI: 10.1109/lsp.2021.3076358. URL: http://dx.doi.org/10.1109/LSP.2021.3076358.

[43] M. Alzantot, Z. Wang a M. Srivastava, *Deep Residual Neural Networks for Audio Spoofing Detection*, čvn. 2019. DOI: 10.48550/arXiv.1907.00501.

[44] S. Scardapane, L. Stoffl, F. Röhrbein a A. Uncini, „On the use of deep recurrent neural networks for detecting audio spoofing attacks“, in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, s. 3483–3490. DOI: 10.1109/IJCNN.2017.7966294.

[45] L. Huang a J. Zhao, „On the Use of LSTM-RNN for Detecting Audio Spoofing Attacks“, in *2022 International Conference on 3D Immersion, Interaction and Multi-sensory Experiences (ICDIIME)*, 2022, s. 147–150. DOI: 10.1109/ICDIIME56946.2022.00040.

[46] Z. Su, „End-to-End Spoofing Speech Detection based on CNN-LSTM“, in *2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 2022, s. 755–758. DOI: 10.1109/ICFTIC57696.2022.10075096.

[47] Z. Teng, Q. Fu, J. White, M. E. Powell a D. C. Schmidt, „ARawNet: A Lightweight Solution for Leveraging Raw Waveforms in Spoof Speech Detection“, in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, s. 692–698. DOI: 10.1109/ICPR56361.2022.9956138.

[48] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi a M. Sahidullah, „ASVspoof 2015: the First Automatic Speaker Verification Spoofing and Countermeasures Challenge“, zář. 2015. DOI: 10.21437/Interspeech.2015-462.

[49] X. Liu, X. Wang, M. Sahidullah et al., „ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild“, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, roč. 31, s. 2507–2522, 2023, ISSN: 2329-9304. DOI: 10.1109/taslp.2023.3285283. URL: http://dx.doi.org/10.1109/TASLP.2023.3285283.

[50] H. Khalid, S. Tariq, M. Kim a S. S. Woo, „FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset“, in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL: https://openreview.net/forum?id=TAXFsg6ZaOl.

[51] R. Reimao a V. Tzerpos, „FoR: A Dataset for Synthetic Speech Detection“, říj. 2019, s. 1–10. DOI: 10.1109/SPED.2019.8906599.

[52] X. Wang a J. Yamagishi, *Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders*, 2023. arXiv: 2210.10570 [eess.AS]. URL: https://arxiv.org/abs/2210.10570.

[53] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar a K. Böttinger, „Does audio deepfake detection generalize?“, *Interspeech*, 2022.

[54] L. Zhang, X. Wang, E. Cooper, N. Evans a J. Yamagishi, „The PartialSpoof Database and Countermeasures for the Detection of Short Fake Speech Segments Embedded in an Utterance“, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, roč. 31, s. 813–825, 2023, ISSN: 2329-9304. DOI: 10.1109/taslp.2022.3233236. URL: http://dx.doi.org/10.1109/TASLP.2022.3233236.

[55] J. Frank a L. Schönherr, *WaveFake: A Data Set to Facilitate Audio Deepfake Detection*, 2021. arXiv: 2111.02813 [cs.LG]. URL: https://arxiv.org/abs/2111.02813.

[56] J. Yi, R. Fu, J. Tao et al., *ADD 2022: the First Audio Deep Synthesis Detection Challenge*, 2024. arXiv: 2202.08433 [cs.SD]. URL: https://arxiv.org/abs/2202.08433.

[57] J. Yi a C. Y. Zhang, *ADD 2023 Challenge Track 1.2 Training/Development Dataset*, Zenodo, čvc. 2024. DOI: 10.5281/zenodo.12151404. URL: https://doi.org/10.5281/zenodo.12151404.

[58] Z. Zhang, Y. Gu, X. Yi a X. Zhao, „FMFCC-A: A Challenging Mandarin Dataset for Synthetic Speech Detection“, in *Digital Forensics and Watermarking: 20th International Workshop, IWDW 2021, Beijing, China, November 20–22, 2021, Revised Selected Papers*, Beijing, China: Springer-Verlag, 2021, s. 117–131, ISBN: 978-3-030-95397-3. DOI: 10.1007/978-3-030-95398-0_9. URL: https://doi.org/10.1007/978-3-030-95398-0_9.

[59] J. Yi, Y. Bai, J. Tao et al., *Half-Truth: A Partially Fake Audio Detection Dataset*, 2023. arXiv: 2104.03617 [cs.SD]. URL: https://arxiv.org/abs/2104.03617.

[60] H. Ma, J. Yi, C. Wang et al., *CFAD: A Chinese Dataset for Fake Audio Detection*, 2023. arXiv: 2207.12308 [cs.SD]. URL: https://arxiv.org/abs/2207.12308.

[61] N. M. Müller, P. Kawa, W. H. Choong et al., „MLAAD: The Multi-Language Audio Anti-Spoofing Dataset“, *International Joint Conference on Neural Networks (IJCNN)*, 2024.

[62] „ASV Spoof Challenge“. (), URL: https://www.asvspoof.org/ (cit. 25. 12. 2024).

[63] Y. Shi, H. Bu, X. Xu, S. Zhang a M. Li, *AISHELL-3: A Multi-speaker Mandarin TTS Corpus and the Baselines*, 2021. arXiv: 2010.11567 [cs.SD]. URL: https://arxiv.org/abs/2010.11567.

[64] J. Yi, C. Y. Zhang, J. Tao et al., *ADD 2023: Towards Audio Deepfake Detection and Analysis in the Wild*, 2024. arXiv: 2408.04967 [eess.AS]. URL: https://arxiv.org/abs/2408.04967.

[65] „Metacentrum NGI“. (), URL: https://www.metacentrum.cz/ (cit. 30. 12. 2024).

[66] „Python - modules“. (), URL: https://wiki.metacentrum.cz/wiki/Python_-_modules (cit. 05. 01. 2025).

[67] J. Černocký, P. Pollák a P. Schwarz. „Česká řečová databáze projektu SpeeCon“. (), URL: https://www.fit.vut.cz/research/product/20/.cs (cit. 04. 01. 2025).

[68] J. Černocký, P. Pollák a P. Schwarz. „Czech SpeechDat(E) Database“. (), URL: https://www.fit.vut.cz/research/product/19/.cs (cit. 04. 01. 2025).

[69] R. Ardila, M. Branson, K. Davis et al., *Common Voice: A Massively-Multilingual Speech Corpus*, 2020. arXiv: 1912.06670 [cs.CL]. URL: https://arxiv.org/abs/1912.06670.

[70] „Transformers“. (), URL: https://huggingface.co/docs/transformers/index (cit. 12. 01. 2025).

[71] „XTTS“. (), URL: https://docs.coqui.ai/en/latest/models/xtts.html (cit. 12. 01. 2025).

[72] „eSpeak NG Text-to-Speech“. (), URL: https://github.com/espeak-ng/espeak-ng (cit. 11. 01. 2025).