BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INTELLIGENT SYSTEMS
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

METHODS FOR REALTIME VOICE DEEPFAKES CREATION

METODY TVORBY HLASOVÝCH DEEPFAKES V REÁLNÉM ČASE

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR KAMBULAT ALAKAEV
AUTOR PRÁCE

SUPERVISOR Mgr. KAMIL MALINKA, Ph.D.
VEDOUCÍ PRÁCE

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

154458

Ústav:	Ústav inteligentních systémů (UITS)
Student:	**Alakaev Kambulat**
Program:	Informační technologie
Název:	**Metody tvorby hlasových deepfakes v reálném čase**
Kategorie:	Bezpečnost
Akademický rok:	2023/24

Zadání:

1. Seznamte se s problematikou deepfakes a metodami pro tvorbu a detekci hlasových deepfakes.
2. Seznamte se parametry, které ovlivňují časovou náročnost metod tvorby kvalitních deepfakes a možnostmi jejich optimalizace.
3. Experimentálně ověřte schopnosti aktuálních nástrojů pro syntézu řeči generovat výstupy v reálném čase. Zvolte nástroje z obou hlavních oblastí - text-to-speech (TTS) a voice conversion (VC).
4. Navrhněte aplikaci, která bude umožňovat tvorbu hlasových deepfakes v téměř reálném čase a bude sloužit jako podpůrný nástroj pro výzkum a pro vzdělávání uživatelů v této oblasti.
5. Navržený nástroj implementujte a proveďte testování funkčnosti a spolehlivosti výsledné implementace. Kvalitu generovaného hlasu ověřte alespoň 2 detekčními metodami. Ideálně ověřte dva scénáře - online i offline přehrání.
6. Realizujte experiment, ve kterém ověříte schopnost lidí rozpoznávat deepfakes tvořený aplikací.
7. Diskutujte, jaké dopady mají dosažené výsledky na bezpečnost současných metod biometrické autentizace.

Literatura:

- FIRC Anton a MALINKA Kamil. The dawn of a text-dependent society: deepfakes as a threat to speech verification systems. In: Brno: Association for Computing Machinery, 2022
- FIRC Anton, MALINKA Kamil a HANÁČEK Petr. Creation and detection of malicious synthetic media - a preliminary survey on deepfakes. In: *Sborník příspevků z 54. konference EurOpen.CZ, 28.5.-1.6.2022*. Radešín, 2022, s. 125-145. ISBN 978-80-86583-34-1.
- PRUDKÝ Daniel, FIRC Anton a MALINKA Kamil. Assessing the Human Ability to Recognize Synthetic Speech in Ordinary Conversation. In: *2023 International Conference of the Biometrics Special Interest Group (BIOSIG)*. Darmstadt, 2023.

Při obhajobě semestrální části projektu je požadováno:
Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Malinka Kamil, Mgr., Ph.D.**
Vedoucí ústavu:	Hanáček Petr, doc. Dr. Ing.
Datum zadání:	1.11.2023
Termín pro odevzdání:	9.5.2024
Datum schválení:	6.11.2023

Abstract

This thesis explores the possibility of achieving real-time voice deepfake generation using open-source tools. Through experiments, it was discovered that the generation rate of voice deepfakes is affected by the computing power of the devices running the speech creation tools. A deep learning model was identified to be capable of generating speech in near real time. However, limitations in the tool containing this model prevented continuous input data for real-time generation. To address this, a program was developed to overcome these limitations. The quality of the generated deepfakes was evaluated using both voice deepfake detection models and human online surveys. The results revealed that while the model could deceive detection models, it was not successful in fooling humans. This research highlights the accessibility of open-source voice synthesis tools and the potential for their misuse by individuals for fraudulent purposes.

Abstrakt

Tato práce zkoumá možnosti generování hlasových deepfake v reálném čase pomocí nástrojů s otevřeným zdrojovým kódem. Experimenty bylo zjištěno, že rychlost generování hlasových deepfakes je ovlivněna výpočetním výkonem zařízení, na kterých jsou nástroje pro tvorbu řeči spuštěny. Byl identifikován model hlubokého učení, který je schopen generovat řeč téměř v reálném čase. Omezení nástroje obsahujícího tento model však bránila kontinuálnímu zadávání vstupních dat pro generování v reálném čase. K řešení tohoto problému byl vyvinut program, který tato omezení překonává. Kvalita generovaných deepfakes byla hodnocena jak pomocí modelů pro detekci hlasových deepfake, tak pomocí online průzkumů na lidech. Výsledky ukázaly, že zatímco model dokázal oklamat detekční modely, nebyl úspěšný při oklamání lidí. Tento výzkum upozorňuje na dostupnost nástrojů pro syntézu hlasu s otevřeným zdrojovým kódem a na možnost jejich zneužití jednotlivci k podvodným účelům.

Keywords

deepfakes, voice deepfakes, biometric systems, realtime voice synthesis, synthetic speech, deep learning, cybersecurity, text-to-speech, voice conversion, open-source deepfake tools, voice deepfake detection

Klíčová slova

deepfakes, hlasové deepfakes, biometrické systémy, syntéza hlasu v reálném čase, syntetická řeč, hluboké učení, kybernetická bezpečnost, převod textu na řeč, konverze hlasu, open-source deepfake nástroje, detekce hlasového deepfake

Reference

ALAKAEV, Kambulat. *Methods for Realtime Voice Deepfakes Creation.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Mgr. Kamil Malinka, Ph.D.

Rozšířený abstrakt

Vzhledem k rychlému pokroku v oblasti hlubokých neuronových sítí jsou stále častěji veřejně dostupné modely schopné řešit různé druhy problémů, které se dříve zdály být extrémně složité. Pokrok v této oblasti má však i svou negativní stránku. Pomocí modelů hlubokého učení je možné vytvářet různé druhy falešných mediálních informací: falešné fotografie, které nikdy neexistovaly, nebo imitace řeči určitých lidí, kteří říkají věci, které nikdy neřekli. To vše lze využít k podvodným nebo provokativním účelům.

Například pomocí hlasových deepfakes může podvodník napodobit hlas skutečné osoby, aby jejím jménem uskutečnil všemožné hovory za účelem vlastního zisku, ať už jde o převod peněz nebo získání osobních informací. Taková osoba však musí mít velmi pokročilý nástroj schopný generovat řeč v reálném čase a v dostatečné kvalitě, aby při telefonování nevzbudila podezření.

Tato práce se zaměřuje na zkoumání schopnosti generovat hlasové deepfakes v reálném čase a hodnocení toho, jak snadné je toho dosáhnout pomocí existujících nástrojů. To vše s cílem posoudit, zda téměř každý, kdo má počítač, internet a volný čas, může získat nástroj pro generování řeči, který lze použít k podvodným účelům.

V rámci práce bylo nutné určit, na čem může záviset rychlost generování řeči. Bylo rozhodnuto zjistit, jak moc závisí rychlost generování na výpočetním výkonu zařízení, na kterém byl nástroj pro generování řeči spuštěn. Experimentální výsledky několika předem natrénovaných modelů převodu textu na řeč a konverze hlasu na různých zařízeních s různým výpočetním výkonem ukázaly velkou korelaci a byl identifikován model s názvem Glow-TTS, který byl schopen generovat řeč během jedné sekundy.

Tento předem natrénovaný model byl k dispozici v konzolovém nástroji Coqui TTS s otevřeným zdrojovým kódem. Ovšem tento nástroj měl některá omezení, která bránila využití modelu v reálném čase. Proto bylo rozhodnuto napsat program, konkrétně uživatelské rozhraní, pod jehož kapotou by běžel nástroj Coqui TTS, a tento program by řešil omezení původního nástroje. Vytvořený program umožňuje používat předem natrénované modely dostupné v nástroji Coqui TTS a odstraňuje omezení bránící generování řeči téměř v reálném čase.

Dále bylo nutné vyhodnotit kvalitu řeči generované modelem Glow-TTS. To znamená vyhodnotit, zda je tento model schopen oklamat metody detekce hlasových deepfakes i skutečné osoby. K vyhodnocení kvality modelu byly vybrány 4 modely detekce syntetické řeči. Podle výsledků experimentu bylo zjištěno, že použité modely s pravděpodobností v rozmezí 80-85% vyhodnotily deepfakes generované modelem jako skutečnou lidskou řeč.

Aby bylo možné provést experiment zaměřený na schopnost člověka určit, zda je hlas na nahrávce imitací řeči, nebo zda patří skutečné osobě, bylo rozhodnuto vytvořit online průzkum, kterého se zúčastnilo 13 osob. Průzkum obsahoval audiosoubory promíchané se skutečnou a umělou řečí a osoba měla určit, které nahrávky jsou deepfakes a které ne. Výsledky průzkumu ukázaly, že lidé s téměř stoprocentní pravděpodobností dokázali rozpoznat deepfakes a skutečnou řeč. Na základě toho lze říci, že použitý model je schopen generovat řeč téměř v reálném čase, stejně jako má dobrou šanci oklamat metody detekce hlasových deepfakes, ale v současné době není schopen oklamat skutečnou osobu.

Na základě provedených experimentů je možné učinit závěr, že rychlý vývoj a dostupnost modelů tvorby hlasových deepfakes může v blízké budoucnosti představovat velmi vážné ohrožení bezpečnosti osobních údajů i peněžních prostředků a také se může stát dalším mocným nástrojem v rukou podvodníků.

Methods for Realtime Voice Deepfakes Creation

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. Mgr. Kamil Malinka, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Kambulat Alakaev
April 30, 2024

Acknowledgements

I would like to express my gratitude for the guidance and advice provided by my supervisor Mgr. Kamil Malinka, PhD. Many thanks to my family who supported me throughout my studies.

Contents

List of Figures

Chapter 1

Introduction

Deepfake is a term used to denote a type of synthetic media created using artificial intelligence(AI) techniques. These techniques allow one to get convincing results and give the impression that the person is saying or doing something that he or she did not actually say or do. Typically, there are two types of deepfakes commonly used: video and audio deepfakes. Aside from the fact that deepfakes are largely used as entertainment today this is not their only use. In terms of cybersecurity, deepfakes are a significant threat for systems and organizations that use video or audio biometrics as part of their security. Instead of relying on traditional methods, such as passwords, biometric systems use unique physical and behavioral attributes to confirm identity.

The motivation for people who use deepfakes to bypass biometric systems is mainly to gain access to personal data or to perform transactions, such as money transfers, by impersonating a real user. The possibility that personal data or money can be used on our behalf throws a shadow over the security provided by institutions. In addition, with the development of machine learning technologies and, consequently, deepfakes, more and more open-source tools are appearing on the Web to create various types of deepfakes that can be used even by people without deep knowledge in this area. Therefore, organizations that use biometric data protection systems are forced to continually improve these systems to successfully resist any kind of fraud or unauthorized access. It is interesting to note that both deepfakes and biometric systems use deep learning technology at their core, and it can be said that their confrontation is a struggle between deep learning models, where typically the better designed and well-trained model wins.

This thesis will focus on voice deepfake technology and on the profile of a person using such technology to fraud and bypass voice biometric systems. Understanding the potential attack vector and the actions of such individuals can help improve the defense and security of biometrics systems for greater preservation of privacy.

To give an example from life, numerous banks around the world are already actively using artificial intelligence-based verification technologies to enhance the security of their customers' accounts. One of the most used approaches in banks' security systems is voice verification to match a registered user. Even with your personal information, a fraudster will not be able to access your account by calling the bank because his voice during the conversation will be verified to match the real user.

There is a considerable amount of open-source or commercial software that is capable of generating the speech of a particular individual given sufficient voice samples. A fraudster using such software could attempt to bypass voice verification of a user's voice in security systems, as in the bank account example. However, to do so, he would not only need enough voice samples of the target person, but would also need to generate synthetic speech fast enough not to raise the suspicions of the bank employee. The question is how this can be achieved.

The main goal of this thesis is to bring the voice deepfake generation rate closer to real time using selected software tools from a fraudster's viewpoint to demonstrate how easy it is to use these programs and to denote the importance of understanding the range of fraudster's capabilities, in order to make voice biometric systems more secure.

This thesis will evaluate the text-to-speech and voice conversion models contained in selected voice deepfake creation tools for their ability to generate speech in real or near-real time. Then, using the model or the tool that has shown the best results in terms of speech generation time, a program will be created that provides a user interface that allows one to continuously input data for the selected model. The quality of the deepfakes produced by the selected model will be evaluated using voice deepfake detection models and an online survey with a group of people. Finally, the impact of the results on the security of current voice biometric methods will be discussed.

Chapter 2

Voice Deepfake Technology

This chapter briefly introduces the history of synthetic voice technology and explains the basic principles of the two most commonly used approaches for creating voice deepfakes: text-to-speech(TTS) and voice conversion(VC). We will also introduce and briefly describe existing commercial and open-source tools.

2.1 A brief history of synthetic voice creation

Researchers have been interested in imitating the human voice with technology for a long time. The first attempts were made in the nineteenth century. People have used mechanical devices to create talking dolls and early sound recording technologies to modify and imitate human speech.

Voice manipulation was further developed in the twentieth century with the invention of analogue tape recorders and signal processing technology. Homer Dudley of Bell Laboratories invented the vocoder [34] in the 1930s, a device that analyzes and synthesizes human speech, opening up new possibilities for voice transformation and editing.

Advancements in information technology and artificial intelligence (AI) have led to the development of high-quality synthetic voices. These advanced technologies, supported by neural networks and machine learning algorithms, make it possible to create highly realistic audio recordings that mimic the voices of real people with impressive accuracy.

The breakthrough occurred in 2016 with the introduction of WaveNet [22], a vocoder developed by DeepMind. WaveNet uses a recurrent neural network (RNN) architecture, a type of deep learning model, to generate highly realistic audio waveforms. This breakthrough allowed researchers to create deepfake voices that were remarkably close to human originals, even capturing subtle nuances such as intonation and timbre.

Advancements in deep learning algorithms and computational power have led to the development of sophisticated voice deepfake techniques. In 2017, Google introduced Tacotron [38], a text-to-speech (TTS) algorithm that synthesizes the prosody of speech, including pitch, rhythm, and intonation. These aspects are crucial to the timbre and identity of the voice.

These developments resulted in the creation of tools such as Deep Voice 2 [2], which could synthesize realistic human voices from text input. Such tools, combined with the availability of large datasets of audio recordings, further legitimized the creation of deepfake voices, making it easier for individuals and groups with minimal technical expertise to create high-quality deepfakes. In the last few years, new techniques have emerged that can generate high-quality voices with a wider range of emotions and accents.

This has expanded the potential applications of deepfake voices beyond creating fake news or impersonation scams. Deepfake voices have the potential to revolutionize fields such as entertainment, education, and healthcare. For example, they could be used to create AI-powered audiobooks, personalize language learning experiences, or assist in speech therapy.

However, the ethical implications of deepfake voices remain a pressing concern. The potential for misuse of technology, such as creating deepfakes to spread misinformation, impersonate public figures, or commit fraud, highlights the need for responsible development and use of this powerful tool. It is important to use technology responsibly to avoid negative consequences.

The increasing complexity of deepfake voices has resulted in several well-known cases where the technology has been exploited for malicious purposes. In 2020, a group of fraudsters used AI-based software to mimic the voice of a CEO and authorize a $35 million money transfer via a phone call [1].

This occurrence, along with other high-profile cases, such as the deepfake video of President Biden, highlights the importance of being vigilant and aware of this emerging technology. Developing robust detection and authentication methods is crucial in countering the spread of deepfakes and their potential to manipulate and deceive.

Regarding voice deepfakes, nowadays there are 2 main approaches to create synthetic speech from the target voice: text-to-speech(TTS) and voice conversion(VC).

2.2 Text-To-Speech Approach

Text-To-Speech (TTS)[31] speech synthesis approach is most commonly used in the field of voice deepfake creation. It is also used in creating voice assistants, audiobooks and educational tools. This technology has many implementation variations, so this section will only describe the basic modern architecture of this approach and the steps of voice synthesis.

TTS - is a natural language modeling process that transforms units of text having a voice sample of the specific person into units of speech for audio representation of that person. It models the natural patterns of human language, including phonetics, prosody, and other linguistic elements.

Basic text-to-speech model consists of several stages:

- Text Analysis
- Linguistic Processing
- Text-to-Phoneme Conversion
- Prosody Modeling
- Acoustic Modeling
- Voice Synthesis

The input text is passed through these stages to obtain the synthetic speech as shown in Figure 2.1.

It is important to note that the architecture can vary significantly depending on the specific approach to implementation. Different TTS implementations may have additional extensions, but the components that are frequently used in many modern text-to-speech systems are presented above. Each stage will be described in more detail below.

An extended model supporting voice generation based on the target voice sample is discussed in Chapter 4.

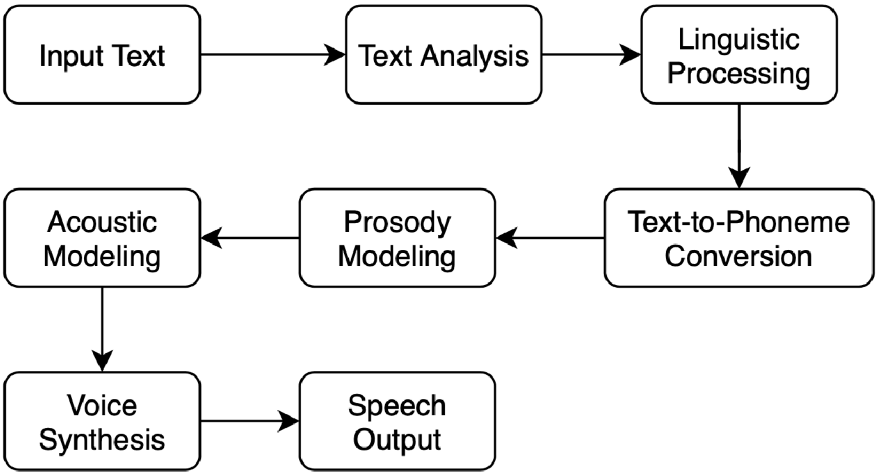


Figure 2.1: Overview of the Text-To-Speech pipeline.

Text Analysis:

This part serves as a preprocessing step where the input text is analyzed to identify linguistic elements such as words, phrases, and punctuation marks.At this stage, the text is broken down into linguistic units. This process is crucial for many Natural Language Processing (NLP) tasks, as it prepares the text for further analysis by machines. By breaking down the text into smaller units, we can more easily identify patterns and relationships between words.

Linguistic Processing:

Linguistic rules and models are used to determine the pronunciation, intonation, and other prosodic features of the text. At this stage, a phonetic representation of words is created. The task of phonetic representation[24] is to depict speech as a physical phenomenon. In essence, it includes measurable properties related to articulation, acoustics and audition.

Text-To-Phoneme Conversion:

Text-to-phoneme conversion(TTP)[4], also called grapheme-to-phoneme conversion (GTP), plays an important role in the naturalness and comprehension of synthetic speech. In text-to-phoneme conversion, words are converted into phonetic transcriptions (phonemes), which are the basic sound units in a language and an accurate representation of pronunciation. In other words, each phoneme corresponds to a particular sound or set of sounds.

There are two main approaches to text-to-phoneme conversion:

- Rule-based conversion: This approach uses a set of rules to map letters (graphemes) to sounds (phonemes). These rules are based on the phonology of the language, which is the system of sounds and sound patterns in a language.

- Statistical conversion: This approach uses statistical models to learn the relationship between letters and sounds. These models are trained on large amounts of data, such as dictionaries and pronunciation recordings.

Prosody Modeling:

To create speech that closely mimics human utterances using text-to-speech technology, it is critical to capture the nuances of speech signals, such as rhythm, intonation, and accent. These elements, called prosody, are not present in text transcripts, but play an important role in transmitting information beyond the textual content. Bringing additional information about prosody into the TTS model is called prosody modeling[27].

Acoustic Modeling:

Acoustic models[3] are used to capture the acoustic characteristics of speech, including the spectral features of different phonemes. By analyzing the spectrum of each phoneme, acoustic models can learn how different sounds are produced by the vocal tract. This information is then used to create synthetic speech that accurately reflects the characteristics of human speech.

In simpler terms, acoustic models are experts at identifying the unique fingerprint of each sound in our language. They analyze how vocal cords, tongue, and mouth combine to create specific frequencies for each phoneme. Using this knowledge in speech synthesis systems, acoustic models help generate speech that sounds natural and very similar to human pronunciation.

Voice Synthesis:

At this step, a text-to-speech system generates the synthetic voice by combining the phonetic, prosodic, and acoustic information. This can be achieved through various synthesis methods such as concatenative synthesis, formant synthesis, and statistical parametric synthesis.

Formant synthesis involves modeling the frequencies of the speech signal, with formants acting as a representation of the resonant frequencies of the vocal tract. These frequencies are estimated during speech synthesis. Articulatory synthesis directly integrates a model of a person's articulatory behavior into the synthesis process. In concatenative synthesis, speech is generated by combining small, pre-recorded speech units to form a complete utterance. The concatenative approach is commonly known as corpus-based speech synthesis. Concatenative synthesis is currently the most widely used approach in text-to-speech (TTS).

2.2.1 Modern text-to-speech tools

In this subsection, we describe several modern commercial and open-source text-to-speech tools in order to choose the most appropriate ones to achieve the goals mentioned in the Introduction 1.

Here's a list of the tools and websites considered:

- Resemble.ai [28] (commercial)
- Play.ht [25] (commercial)
- MBROLA [21] (open-source)
- Coqui TTS [10] (open-source)
- Real Time Voice Cloning(RTVC) [13] (open-source)

Resemble.ai offers features for generating high-quality, realistic sounding speech from written text. They also offer an extensive library of AI voices in many languages, allowing one to choose the voice with which to generate speech.

While the free version of the tool is limited in terms of the available functionality for speech creation, the paid version allows one to create a limited number of digital copies of the provided voice (your own or another person's), on the basis of which the speech synthesis will be performed. However, for security reasons, Resemble.ai does not allow one to simply duplicate a voice from an audio recording and generate a speech based on it. This requires the voice that the fraudster wants to clone to say a certain phrase and send it as an audio file to Resemble.ai for comparison with the voice from the first audio recording, which makes the use of this tool very inconvenient from the fraudster's viewpoint.

Also, the voice-generating process itself has a disadvantage from the perspective of creating a real-time deepfake: during the generating of the deepfake and until the moment of its full playback, the user has no option to enter another text and has to wait for the end of the generated voice playback. From a fraudster's point of view, this disadvantage can have a significant impact on arousing suspicion during a phone conversation because of the relatively long pauses in the conversation due to the need to enter a new text and then generate a deepfake. All of this makes this tool less attractive from the fraudster's perspective.

Play.ht offers rich functionality for the text-to-speech(TTS) service that focuses on providing high-quality speech using a vast library of AI-generated voices. Play.ht provides one of the most extensive libraries of AI voices available, with nearly 600 voices across over 142 languages and accents.

It also allows one to clone the voice of a real person and generate synthetic speech with it, but just like text-to-speech from Resemble.ai, it does not allow one to enter other text during speech generation. More precisely, trying to enter new text immediately stops the generated deepfake for the previous text from playing.

From a fraudster's perspective, the use of commercial tools can lead to lower anonymity due to the need to register account and provide personal information when paying for a subscription in order to receive a wider range of tools and less limitations. In addition to the disadvantages described above, the user does not have the ability to make modifications to commercial tools, making them less flexible to use. Based on this, it was decided to focus on open-source TTS tools.

MBROLA is an open-source speech synthesizer that does not directly accept raw text as input. Instead, it relies on pre-recorded speech samples called „diphones." But to obtain a full TTS system, one needs to use this synthesizer in combination with a text processing system that produces phonetic and prosodic text representation. It complicates the work with this tool and, in addition, has no support for the option to generate speech based on your own voice.

Real Time Voice Cloning(RTVC) is an open-source program that provides a simple GUI to synthesize speech using pre-trained models. Real Time Voice Cloning is one of the most popular and commonly used TTS voice deepfake creation tools. Its main feature is the ability to synthesize speech based on short embedding recordings. This makes it highly flexible as the pre-trained models are independent of the target speaker. Since unlike commercial solutions, this tool can be installed on your device for more detailed testing, and unlike MBROLA, it is a complete platform for speech creation, it was decided to use this tool for experiments. This tool is discussed in more detail in Chapter 4.

Coqui TTS is a powerful open-source toolkit mainly designed for text-to-speech (TTS) tasks. Essentially, the tool is a wide set of pre-trained models designed using text-to-speech technology. With the API and the availability of large amounts of pre-trained models, this tool is very flexible to customize and modify, allowing one to conduct a wide range of experiments. This tool is discussed in more detail in Chapter 4.

2.3 Voice Conversion Approach

Voice conversion [33] is a technique used to modify the specific voice of a source speaker to match the vocal quality of a target speaker. Unlike TTS, this process is independent of speech content and, therefore, does not require transcription. Some state-of-the-art voice conversion frameworks can individually transmit speech components such as timbre, pitch, or rhythm. Voice conversion is most common in online games or voice imitation and remix songs. Voice conversion tools can change any vocal characteristics of the original voice (age, gender, etc.) to hide a person's true identity.

A typical voice conversion pipeline consists of speech analysis, mapping and reconstruction modules, as shown in Figure 2.2, which is called the analysis-mapping-reconstruction pipeline. The speech analyzer decodes the speech signals of the original speaker into features representing suprasegmental and segmental information [18], and the mapping module adapts them to the target speaker, while the reconstruction module re-synthesises the speech signals in the time domain. Suprasegmental features cover aspects of speech, such as pitch, rhythm, and stress, that extend beyond individual sounds. Segmental features, on the other hand, refer to individual sounds or segments of speech. The mapping module plays a key role in many studies. These techniques can be classified in different ways, for example, according to the way in which they use training data.

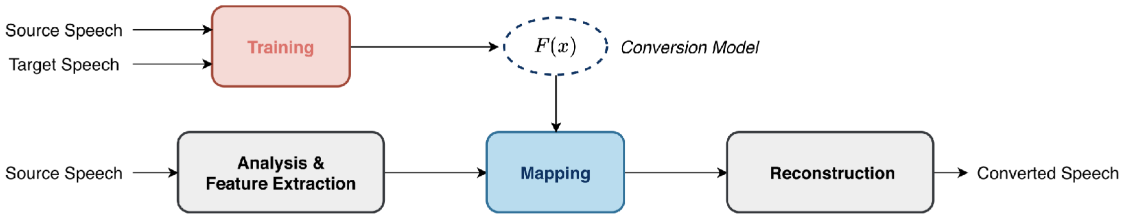


Figure 2.2: Flow of a voice conversion system. Image is adapted from [33].

There are two major approaches to voice conversion: parallel and non-parallel. Parallel and non-parallel voice conversion methods differ in training data requirements.

In parallel voice conversion, the system is trained using paired datasets, where each utterance of the source speaker (whose voice is being modified) has a corresponding equivalent uttered by the target speaker (the desired voice). This one-to-one pair serves as a powerful teacher of the model. It can directly analyze the specific acoustic transformations required to transfer the characteristics of the source speaker to the target speaker. This can be thought of as learning a new accent - by comparing one's own pronunciation of words with recordings of a native speaker, one can identify subtle differences in pitch, timbre, and articulation. The parallel voice transformation takes this analogy one step further. By directly comparing the corresponding segments of source and target speech, the model goes beyond the general characteristics. It learns details that distinguish the two speakers, such as the way certain sounds are pronounced or the rhythm of their speech. This deep learning process allows parallel methods to achieve remarkably high accuracy in speech transformation.

Nonparallel voice conversion, on the other hand, does not require a one-to-one mapping between source and target speakers during training. Instead, the model learns to generalize between different speakers using a pool of speakers. This method is more flexible and does not require a perfectly aligned data set, which makes it useful in scenarios where precisely aligned data are difficult to obtain. However, nonparallel methods may have problems capturing the specific nuances of individual speakers, which means that they may not capture the unique characteristics of a particular individual as accurately as a parallel approach. Imagine trying to mimic a friend's laughter by listening to a group of laughing people - you might pick up the general sound, but you probably will not notice the specific characteristics and subtle changes that make their laughter instantly recognizable. Similarly, non-parallel conversion can create a voice that falls into a target category (for example, a baritone or youthful voice), but it will not perfectly capture the specific voice usage of a particular actor or person in that category.

Both methods aim to modify the voice of the source speaker to make it similar to that of the target speaker, which allows their use in applications such as voice imitation, identity masking or generating different speech outcomes in different sound scenarios. The choice between parallel and non-parallel approaches often depends on the availability and type of training data, as well as on the specific needs of the voice conversion application.

2.3.1 Modern voice conversion tools

Tools of this type are less common than text-to-speech tools. Voice conversion is used mostly as entertainment or to voice characters in video games, allowing one to change the parameters of speech. For example, to sound like a robot. However, there are a limited number of tools available in the public domain that can generate speech based on a real voice. The author selected 2 voice-conversion tools:

- Respeecher [30] (commercial)
- FreeVC [19] (open-source)

Respeecher is a commercial voice conversion tool known for its ability to create realistic voice clones. The tool has a wide base of available voices to use as a target voice that pronounces text from the source voice. However, in order to add a new voice that is not among the available ones, user needs to submit a request for approval to the admins of the tool, which may cause difficulties from the fraudster's perspective.

FreeVC is an open-source voice conversion model that allows one to change the source speaker's voice to sound as the target speaker's voice without changing the content of the speech. This model is pre-trained and contained in the Coqui TTS tool mentioned earlier, which makes it possible to use and test this model in a more flexible way. The model is discussed in more detail in Chapter 4.

Chapter 3

Preface to the following chapters

3.1 Core ideas of this paper

This bachelor thesis was originally started by the author as a project as part of the subject „Project Practice“. The reason for this thesis is to investigate how available voice generation tools can be used by fraudsters for personal gain, such as accessing people's personal information by deceiving their relatives and friends, or attempting to convince them to transfer money using another person's voice. In other words, the purpose is to find out whether such technologies and their availability to the public today pose a risk to the security of people's personal data and their financial resources.

Certainly, to successfully perform this type of fraud, the fraudster will need a modern tool capable of generating high-quality speech in real or near-real time. In order to determine how much existing software tools are capable of generating voice deepfakes in real or fairly close to real time, it was decided to try to answer two questions that appeared to the author while researching the relevant literature for this project: „On what parameters can the rate of voice deepfakes generation depend?“ and „Is it possible to achieve real-time voice generation rate by changing these parameters?“.

The author of this thesis assumed that there may be a dependence of the speech synthesis rate time on the computing device on which the software tool was run. This assumption was based on the fact that deepfakes today are mostly created using deep neural networks, which, in their turn, require certain computing power for their work. Chapter 4 is focused on checking the dependence of the voice deepfake synthesis rate on the hardware where the voice creation tool was run.

The author also supposed that depending on the software tool, it is possible to try to optimize the program to achieve higher performance of the software, which, however, would require serious skills in programming and advanced knowledge in AI and deep neural networks. Some kind of optimization of one of the tools used without changing its underlying code is discussed in Chapter 5.

When searching for answers to the above-mentioned questions, the author intended to identify the most appropriate software tools or deep learning models used in this thesis to generate deep voices in near real time. Given that the author puts himself in the shoes of a fraudster who wants to use voice generation technology for illegal purposes, it is also necessary to evaluate the quality of speech generated by the selected software tool. The evaluation of the quality of the generated deepfakes is conducted and discussed in Chapter 6.

Based on the experiments conducted, the author discusses in Chapter 7 the threat that speech synthesis technology that is publicly available today may pose to voice biometric systems and ordinary people.

Chapter 4

The ability of speech synthesis tools to generate real-time voice deepfakes

This chapter performs experiments in order to examine the ability to create real-time voice deepfakes based on existing open-source programs. Experiments and measurements are performed on the basis of the hypotheses of a subchapter 4.1 to confirm them.

4.1 Determining the approach and actions in conducting experiments

Within the scope of this chapter, it is necessary to reveal the presence or absence of dependence of time of creation of voice deepfakes with the use of the selected program tools on the performance of the device, on which the program was executed. The assumption about the dependence of deepfakes creation time on the device (computer) performance is based on the fact that such programs use deep neural networks and their applications, the time of data processing in which, as is known, depends on the performance of the computing machine. Thus, the goal of this chapter is to confirm this assumption and experimentally find out if only by changing the performance of the computing device it is possible to bring ready-to-use voice synthesis models to real-time creation rate for the text-to-speech (TTS) or voice conversion (VC) models.

This chapter contains three different experiments, each conducted as follows: we take one of the tools selected in Chapter 2 and test whether there is a dependence of the speech synthesis time on the power of the computer where the tool was run. It is also evaluated whether it is possible to achieve real-time speech synthesis using the selected tool by only increasing the computing power of the device.

4.2 Experiments with Real Time Voice Cloning tool

While commercial tools provided the most quality synthetic speech, their usage is limited and one does not have any possibility to try to accelerate time of the speech generating by changing the device configuration, because all computations are performed remotely and of course one does not have access to the source code of such tools. Thus, the only solution is to use open-source tools that could also be installed on the device. For the goals of this experiment, the open-source tool Real Time Voice Cloning was selected.

It is based on the implementation of the SV2TTS [14] framework. The SV2TTS framework has three stages:

- A speaker encoder that extracts embeddings from a single speaker's short utterance. Embedding is a representation of a particular person's voice such that similar voices are close in latent (vector) space. It can be said that embedding captures what the speaker sounds like.

- A synthesizer that, on the basis of the fact that a particular speaker is embedded in the text, generates a spectrogram from the text.

- A vocoder that outputs an audio signal (waveform) from a spectrogram generated by a synthesizer. The speaker vocoder receives a short utterance of the speaker to clone. It generates an embedding, which is used to control the synthesizer, and the text, treated as a sequence of phonemes, is sent to the input of the synthesizer. The synthesizer takes a sequence of phonemes, which are the smallest units of human sound, and embeddings from the encoder, and uses the Tacatron 2 [32] architecture to generate frames of mel-spectrograms. The vocoder takes the output of the synthesizer to generate the speech waveform. This is illustrated in Figure 4.1.

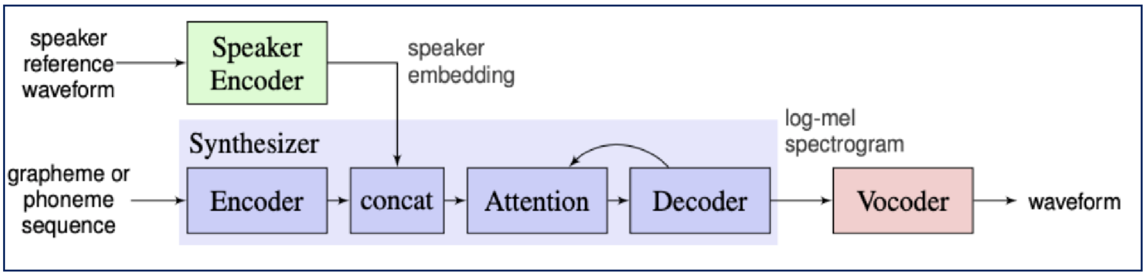


Figure 4.1: The SV2TTS overview. Each of the three components are trained independently. Figure is extracted from [14].

Although all parts of the system are trained separately, there is still a requirement that the synthesizer has embeddings from the trained encoder and the vocoder has mel-spectrograms from the trained synthesizer. Figure 4.2 illustrates how each part of the framework depends on the previous one for training. The speaker encoder must be generalized enough to create meaningful embeddings on the dataset of the synthesizer.

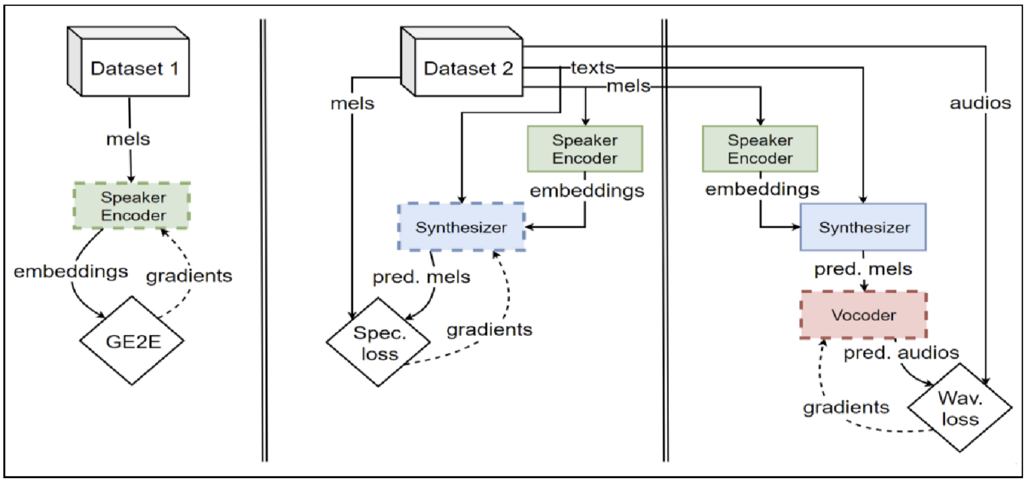


Figure 4.2: The sequential three-stage training of SV2TTS. Models with solid contour lines are pre-trained. Figure is extracted from [13].

As stated in the purpose of the experiment, in order to define how to accelerate the creation of voice deepfakes, the experiment will study if there is any correlation between the inference rate of the voice deepfake creation and the configuration of the device. To begin with, it should be said that the author of this thesis has not found any research dedicated to this issue in public access, regardless of the software tool in question. Thus, the research will be performed on the basis of the available information within the scope of the issue and the author's assumptions.

Firstly, it is necessary to determine which RTVC tool modules are most likely to depend on the PC configuration. RTVC authors reproduced the encoder model with their own PyTorch implementation. The RTVC authors determined through experiment that the encoder is by far the fastest of the three models, as it operates at approximately 1000 × real time by testing it on NVIDIA GTX 1080 GPU. The synthesizer is Tacotron 2 without Wavenet. RTVC authors used an open-source TensorFlow implementation of Tacotron2 from which they strip Wavenet and implement the modifications added by SV2TTS.

In SV2TTS and in Tacotron2, WaveNet is the vocoder. WaveNet open-source version is also known to be the slowest practical deep learning architecture at inference time. Since wavenet is a deep neural network with many layers, the author of this thesis supposes that wavenet is highly dependent on the configuration of the running device, especially on GPU. The RTVC authors made no changes to the vocoder. With the above information, it can be preliminarily assumed that the encoder, synthesizer, and vocoder are probably dependent on the PC configuration, since the libraries used in their implementation are directly dependent on the computing power of the PC and concretely on the CPU and GPU.

In order to assess whether there is a correlation, the author has prepared four different performance computers to be used to measure the speed of creating voice deepfakes. PyTorch, which is required to run the RTVC tool, provides a choice of computing platform. The three devices used in this experiment will use the GPU platform and one will use the CPU platform (due to technical limitations).

The configurations of the devices and compute platforms are shown in Table 4.1.

Name	**CPU**	**RAM**	**GPU**	**PyTorch platform**
MacBook Pro 13 (2017)	Dual-Core Intel Core i5 (2.3 GHz)	8 GB	Intel Iris Plus Graphics 640 1536 MB	CPU
Acer Nitro 5 AN515-54-50RC	Intel Core i5-8300H (2.3 GHz)	8 GB	NVIDIA GeForce GTX 1050 3GB	GPU
Lenovo Legion Y540-15IRH	Intel Core i7-9750HF (2.6 GHz)	16 GB	NVIDIA GeForce GTX 1660 Ti 6 GB	GPU
Galdor Metacentrum (CESNET)	AMD EPYC 7543 (2.8GHz)	512 GB	NVIDIA A40 48 GB	GPU

Table 4.1: Device configuration and used PyTorch computing platforms.

The vocoder selection parameter has an option to select the Griffin-Lim algorithm instead of the default (WaveNet) model. The Griffin-Lim algorithm is not a machine learning model, but it also produces a fairly good output. However, the synthetic voice may sound less natural than the voice produced by the default model. For this reason, the default model will be used as a vocoder. In order to create the same conditions for the experiment on all computers, all other RTVC parameters are also set to default.

The RTVC authors combined several noisy datasets to make for a large corpus of speech of quality similar to what is found in the wild. These datasets are LibriSpeech [23], VoxCeleb1 [20], VoxCeleb2 [9] and an internal dataset. LibriSpeech is a corpus of audiobooks that makes up 1000 hours of audio from 2400 speakers, split equally into two sets of 'clean' and 'other'. For the goals of this experiment, the LibriSpeech/train-clean-100 [35] dataset is used. This dataset contains 100 hours of 'clean' English speech.

As the authors of the RTVC point out, if the utterance that was given as input is shorter than 12.5 seconds, then the model will run slower than real time. For this reason, only phrases longer than 12 seconds will be used in the experiment to create the same conditions on all devices. From the integrated RTVC test dataset, 10 utterances in the male voice and 10 utterances in the female voice were selected. Each utterance is longer than 12 seconds. The inference time will be measured on each computer for male and female utterances separately and as one set of utterances. It will also take into account the output time at a specified length of target text (what the synthesized voice should say). A short target text will be marked as less than or equal to 15 words, and a long target text will be marked as more than 30 words.

The results of the synthetic voice generation rate based on the above statements for MacBook Pro 13 (2017) are shown in Table 4.2.

	Male utterances	Female utterances	Male and Female utterances
Short target text	29 sec.	29 sec.	28 sec.
Long target text	45 sec.	56 sec.	47 sec.

Table 4.2: Synthetic speech creation time on MacBook Pro 13 (2017).

From the above information, it can be assumed that the creation time of deepfake also depends on the length of the target text. To confirm this assumption, it needs to be proved on the tests of the other computers as well. The results of the synthetic voice generation for Acer Nitro 5 are shown in Table 4.3.

	Male utterances	Female utterances	Male and Female utterances
Short target text	21 sec.	20 sec.	23 sec.
Long target text	17 sec.	17 sec.	19 sec.

Table 4.3: Synthetic speech creation time on Acer Nitro 5.

It can be noted that on a more performance computer configuration, speech generation is 2 and sometimes 3 times faster than on the previous one.

The results of the synthetic voice generation for Lenovo Legion are shown in Table 4.4.

	Male utterances	Female utterances	Male and Female utterances
Short target text	11 sec.	17 sec.	13 sec.
Long target text	12 sec.	13 sec.	12 sec.

Table 4.4: Synthetic speech creation time on Lenovo Legion.

From the data from Table 4.4, it can be seen that the time rate of synthetic speech creation has increased significantly compared to the results from Table 4.2. Also, from the table above it can be seen that not only there is almost no difference in speech generation rate for long and short target texts for this configuration, but also the speech generation for long target texts can be completed faster than for short ones. The RTVC authors also state that speech deepfake creation is faster for longer target texts than for shorter ones. However, considering the results of Table 4.2, author of this thesis assumes that this statement is true only for high-performance computers.

The results of synthetic voice generation for the powerful MetaCentrum cluster „Galdor“ are shown in Table 4.5. MetaCentrum is a virtual service providing the opportunity to use high-performance computing hardware.

	Male utterances	**Female utterances**	**Male and Female utterances**
Short target text	7.6 sec.	7.5 sec.	8 sec.
Long target text	8.4 sec.	8.2 sec.	7.5 sec.

Table 4.5: Synthetic speech creation time on Galdor (CESNET).

With this configuration, it can be seen that the time rate of deepfake creation has increased compared to the results of Table 4.4, but not so significantly, provided that the difference in device performance is large. The results of the experiment show that even with a very high-performance configuration, it was unsuccessful to achieve near-real-time deepfake creation rate.

The experiment confirmed the presence of a significant correlation between the rate of generation of voice deepfakes and the configuration of the computer on which the speech generation software was running. During the experiment, the significant dependence of deepfake output rate on the length of the target sentence was found, but it is true only for low-performance configurations.

The dependence of the voice deepfakes generation time rate on the performance of a computing device is shown in Figure 4.3. The x-axis shows the devices themselves, and the y-axis shows the time rate of deepfakes generation in seconds. Measurements are shown for target texts of different lengths.

Testing of the output rate on a powerful computing configuration of the Galdor cluster(Metacentrum) did not show results close to real time, which proves that only by improving the device configuration, it is not possible to achieve the desired inference rate.

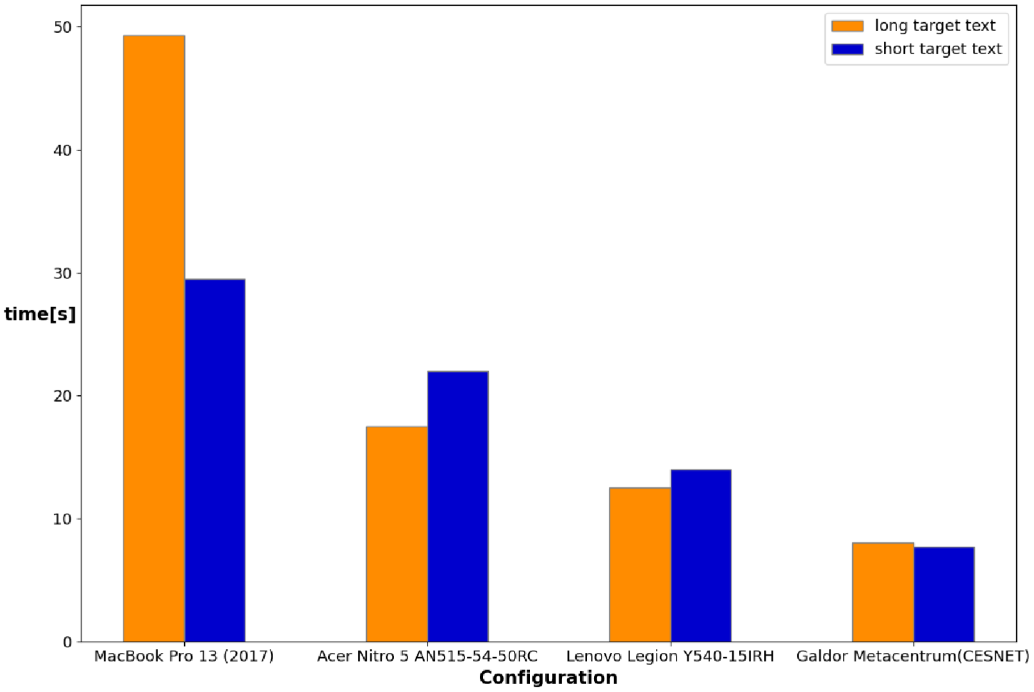


Figure 4.3: Dependence of voice deepfake generation time rate on the performance of a computing device.

Thus, the only way to improve the inference time rate of deepfake generating using Real Time Voice Cloning program is to try to manipulate the source code of the program and the architecture of the models responsible for the stages of voice deepfake creation. However, due to the high entry threshold to perform such manipulations and also considering that this work assumes only basic programming skills of a person who wants to achieve real-time generation rate for their own purposes (legitimate or not), these manipulations will not be performed. Instead, a similar experiment as in this subchapter will be conducted using a different, more modern, and variant software tool with the option to choose from several TTS models and more flexible in terms of source code tuning.

4.3 Experiments with Coqui TTS tool

Coqui TTS is an open-source toolkit for building text-to-speech systems. It is developed by Coqui AI, a community-driven initiative focused on providing accessible and customizable machine learning tools. Coqui TTS is built on the Tacotron2 and WaveGlow [26] architectures, which are deep learning models used for text-to-speech synthesis. It provides pre-trained models for multiple languages, allowing one to convert written text into natural-sounding speech.

Some key features and capabilities of Coqui TTS include:

- Neural text-to-speech synthesis: Coqui TTS utilizes neural network models to convert text into speech, offering more natural and expressive output.
- Multilingual support: Coqui TTS supports various languages, with pre- trained models available for languages like English, Spanish, French, German, and more.
- Open-source and customizable: Being an open-source toolkit, Coqui TTS allows one to access and modify the underlying code and models, enabling customization and adaptation to specific use cases.
- Integration with Python: Coqui TTS provides Python bindings, making it easy to use within Python applications and workflows.

This experiment will evaluate the rate at which pre-trained models from the tool above generate deepfakes.

For the experiment, the most high-performance models in terms of the ratio of performance and quality of the generated deepfake were selected.

Models Used: Glow-TTS [15], VITS [16], YourTTS [5].

Glow-TTS is a text-to-speech (TTS) model that belongs to the family of generative flow-based models. It is designed to convert written text into natural-sounding speech. The Glow-TTS model utilizes a combination of neural network architectures, including a modified WaveNet vocoder. The model takes linguistic features as input, which encode the linguistic content of the text, and generates a sequence of acoustic features, which represent the speech waveform. The key idea behind Glow-TTS is to model the probability distribution of the acoustic features conditioned on the linguistic features using a flow-based approach. The advantage of the Glow-TTS model is its ability to generate speech with relatively fewer model parameters compared to other TTS models, which can lead to faster synthesis and lower resource requirements.

YourTTS is a text-to-speech (TTS) model and it stands out from the rest models that require vast amounts of speaker-specific recordings, as it provides impressive voice cloning with minimal data („few-shot“ learning approach). This is made possible through a combination of two key techniques:

- Speaker embedding layer: This component acts as a voice fingerprint scanner. During training, YourTTS receives multiple speech samples and extracts the essence of each speaker's voice, capturing unique characteristics such as pitch, timbre, and speaking style.

- Domain-specific adaptation: Fine-tunes the model on a small amount of target speaker data. This process clarifies the model's understanding of the target speaker's voice, allowing it to synthesize speech that is very similar to that speaker.

The model can synthesize speech for unseen speakers based on their textual descriptions (e.g., gender, age, accent) without prior training data for that specific voice.

VITS is a text-to-speech (TTS) model that combines a conditional variational autoencoder (VAE) with adversarial learning techniques to generate speech waveforms from input text. The model architecture is a combination of the GlowTTS encoder and the HiFiGAN [17] vocoder.

The VITS model follows an end-to-end approach, which means that it directly converts text into synthesized speech without relying on separate components for text analysis, linguistic features, or acoustic modeling. To enhance the quality and naturalness of synthesized speech, VITS incorporates adversarial learning using a discriminator network. The discriminator is trained to distinguish between real and synthesized mel-spectrograms, and its feedback is used to guide the training of the variational autoencoder(VAE).

The used models were pre-trained by the Coqui team on different datasets. The Glow-TTS model was trained on the LJ Speech Dataset [12]. The provided dataset is a collection of 13,100 short audio clips from a single speaker. These clips are recordings of passages taken from 7 nonfiction books. Each audio clip is accompanied by a transcription. The duration of the clips ranges from 1 to 10 seconds.

The VITS model was used in two versions that are available in Coqui TTS. One version was trained on the VCTK [36] dataset. It is a multilingual dataset that includes speech data uttered by 109 native English speakers with various accents. Each speaker in the dataset reads approximately 400 sentences, resulting in a substantial amount of training data for the VITS model. Another version was trained on the LJ Speech Dataset as well as the Glow-TTS model.

The configurations of the devices for the experiment are shown in Table 4.6.

Name	CPU	RAM	GPU
MacBook Pro 13 (2017)	Dual-Core Intel Core i5 (2.3 GHz)	8 GB	Intel Iris Plus Graphics 640 1536 MB
Acer Nitro 5 AN515-54-50RC	Intel Core i5-8300H (2.3 GHz)	8 GB	NVIDIA GeForce GTX 1050 3GB
Lenovo Legion Y540-15IRH	Intel Core i7-9750HF (2.6 GHz)	16 GB	NVIDIA GeForce GTX 1660 Ti 6 GB
Asus ROG Strix G15	AMD Ryzen 7 4800H (2.9GHz)	16 GB	NVIDIA GeForce RTX 3060 6 GB

Table 4.6: Device configuration.

The inference time will be measured separately on the basis of the length of target text (what the synthesized voice should say) as in the previous experiment. A short target text will be marked as less than or equal to 15 words, and a long target text will be marked as more than or equal to 40 words. Each measurement is the mean value of the deepfake creation rate calculated on the basis of five runs of each model on each configuration.

The results of the synthetic voice generation rate using the VITS model trained with the VCTK dataset are shown in Table 4.7.

Name	Short target text	Long target text
MacBook Pro 13 (2017)	2.75 sec.	8.55 sec.
Acer Nitro 5 AN515-54-50RC	2 sec.	6.82 sec.
Asus ROG Strix G15	1.38 sec.	4.34 sec.
Lenovo Legion Y540-15IRH	1.25 sec.	4.2 sec.

Table 4.7: Synthetic speech creation time using the VITS model trained on the VCTK dataset.

It can be seen that the best result obtained in this test for the VITS model on the VCTK dataset in terms of the time required to generate a voice deepfake is 1.25 seconds for a short target text. This is almost 6 times faster than the best result from the previous experiment, which was 7.5 seconds. However, such a result, although close to real time, is still not satisfactory.

The results of the synthetic voice generation rate using the VITS model trained on the LJ Speech dataset are shown in Table 4.8.

Name	**Short target text**	**Long target text**
MacBook Pro 13 (2017)	3.7 sec.	11.5 sec.
Acer Nitro 5 AN515-54-50RC	2.47 sec.	10.5 sec.
Asus ROG Strix G15	2.1 sec.	7 sec.
Lenovo Legion Y540-15IRH	2 sec.	6.8 sec.

Table 4.8: Synthetic speech creation time using the VITS model trained on the LJ Speech dataset.

It can be seen that for this dataset the model shows worse results than for the previous one, which is the reason why this model trained on this dataset will not be used further within this thesis.

The results of the synthetic voice generation rate using the YourTTS model are shown in Table 4.9. The configurations of the devices used for this experiment are shown in Table 4.11.

Name	**Short target text**	**Long target text**
MacBook Pro 13 (2017)	4.5 sec.	7 sec.
Asus TUF Gaming F15	2.8 sec.	5.4 sec.
Lenovo Legion Y540-15IRH	1.7 sec.	4.2 sec.
Galdor (Metacentrum)	9.5 sec.	12.8 sec.

Table 4.9: Synthetic speech creation time using the YourTTS model.

We can see that the model, despite its advantages, generates speech quite slowly, which is not suitable for our purposes. The author of this thesis did not find an explanation for the significant slowdown in the speech generation rate on the Galdor computing cluster.

The results of the synthetic voice generation rate using the Glow-TTS model trained with the LJ Speech dataset are shown in Table 4.10.

Name	**Short target text**	**Long target text**
MacBook Pro 13 (2017)	0.76 sec.	2.37 sec.
Acer Nitro 5 AN515-54-50RC	0.56 sec.	1.91 sec.
Asus ROG Strix G15	0.43 sec.	1.3 sec.
Lenovo Legion Y540-15IRH	0.36 sec.	1.2 sec.

Table 4.10: Synthetic speech creation time using the Glow-TTS model trained on the LJ Speech dataset.

From the table above it can be seen that the Glow-TTS model shows the best results in terms of the rate of generating voice deepfakes and confirms the hypothesis presented at the beginning of this chapter that by using high-performance hardware on which the software for generating voice deepfakes will be run, it is possible to achieve inference time close to real time.

However, this statement is only true for the TTS models, in the next subchapter a similar experiment will be conducted but for the voice conversion (VC) model.

4.4 Experiments with the Coqui TTS tool using the voice conversion model

In this subchapter we will conduct an experiment with measuring the voice deepfake generation time as in the previous two subchapters, but this time we will test the voice conversion model called FreeVC, instead of the TTS models. This model, as well as the models from the previous subchapter, is one of the pre-trained and the only currently available VC model in Coqui TTS that we will use in this experiment.

FreeVC is a voice conversion (VC) model that can convert the voice of a source speaker to a target speaker voice without the need for text annotation. FreeVC uses an end-to-end neural network architecture that consists of three main components:

- The first component of the FreeVC model is the prior encoder. Its role is to extract clear content information from the source speech signal. This is achieved using a waveform language model (WavLM) [6] to learn the statistical properties of speech sounds. The WavLM is trained on a large corpus of unannotated speech data, so it does not require any text information to learn its representations.

The WavLM converts the input speech signal into mel-spectrograms. These representations capture the frequency content of the signal. The WavLM then uses these mel-spectrograms to learn a probability distribution over a set of latent variables representing the content of the speech signal.

The mel-spectrograms and learned probability distribution are input into the prior encoder, which outputs a set of latent variables representing the speech signal's content. These variables are then passed to the speaker encoder.

- The speaker encoder is the second component of the FreeVC model and is responsible for extracting speaker information from the target voice waveform. It uses a neural network to learn the unique characteristics of the speaker's voice, such as their vocal tract length, formant frequencies, and timbre.

  The mel-spectrograms and content latent variables are input, and a set of speaker latent variables that represent the speaker information are output. These latent variables from the speaker are then passed on to the decoder.

- The decoder is the third and final component of the FreeVC model and is responsible for reconstructing the waveform of the target voice from the extracted content and speaker information. This is done using a WaveNet-style neural network that generates the waveform one sample at a time.

  The WaveNet-style decoder takes the content latent variables and the speaker latent variables as input and outputs a sequence of samples that represent the waveform of the target voice. The decoder then uses a post-processing step to improve the quality of the waveform.

The architecture described above is shown in Figure 4.4.

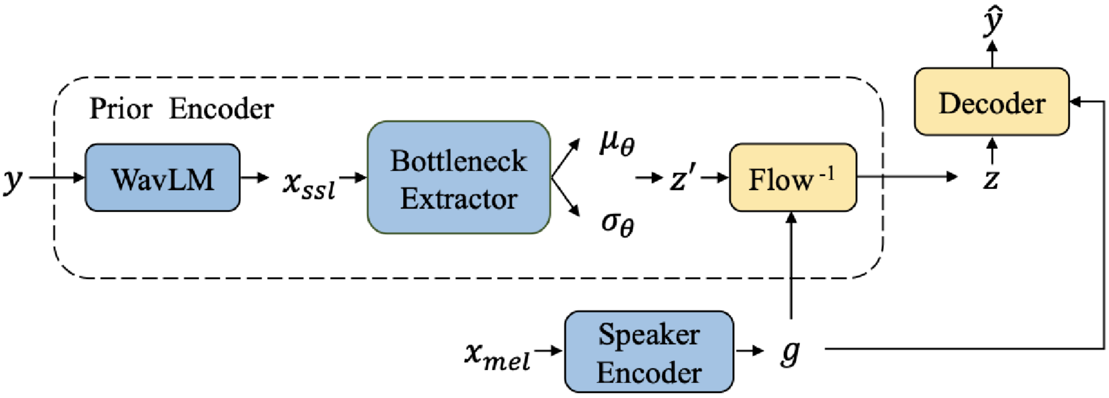


Figure 4.4: Inference procedure. Image is adapted from [19].

The experiment will be performed with an audio file containing the source voice and speech with a duration of 10 seconds and an audio file containing the target's voice with a duration of 1 minute. The results on each device (computer) will be measured 3 times and as a result of each measurement the average value of deepfake generating rate will be taken.

The configurations of the devices for experiment are shown in Table 4.11.

Name	CPU	RAM	GPU
MacBook Pro 13 (2017)	Dual-Core Intel Core i5 (2.3 GHz)	8 GB	Intel Iris Plus Graphics 640 1536 MB
Lenovo Legion Y540-15IRH	Intel Core i7-9750HF (2.6 GHz)	16 GB	NVIDIA GeForce GTX 1660 Ti 6 GB
ASUS TUF Gaming F15	Intel Core i5-10300H (2.5 GHz)	16 GB	NVIDIA GeForce GTX 1650 4 GB
Galdor (Metacentrum)	AMD EPYC 7543 (2.8GHz)	512 GB	NVIDIA A40 48 GB

Table 4.11: Device configuration.

The results of experiment are shown in Table 4.12.

Name	Inference time rate
MacBook Pro 13 (2017)	36.5 sec.
ASUS TUF Gaming F15	27.5 sec.
Lenovo Legion Y540-15IRH	24.2 sec.
Galdor (Metacentrum)	18 sec.

Table 4.12: Synthetic speech creation time using the FreeVC model.

It can be seen that even with the use of a powerful computational cluster, it was not possible to achieve an inference time close to the real time for the FreeVC voice conversion model. In addition, the input data format that involves operating with audio files is not suitable for real-time speech synthesis, as it does not allow for continuous voice input, but only with audio files containing the target text. Thus, the author of this thesis will focus on the text-to-speech models from the previous subchapter.

4.5 Chapter conclusion

The experiments conducted in this chapter using software tools for generating synthesized speech on several devices have shown a fairly strong dependence of the time required to produce a voice deepfake by the program on the device configurations where it was run. It was also proven that even with large increases in device performance, at a certain point, the speech generation rate will stop improving significantly. Based on the experiments of text-to-speech and voice conversion models, it was decided that the use of voice conversion models is not appropriate for the purposes of this thesis, since the output generation rate is very high and the input data format is not suitable for real-time use.

On the basis of the experiments, the Glow-TTS model stands out. Its ability to generate speech in less than a second, even on a low-performance computer, made it the most preferred choice. Its rate is especially important in real-time scenarios, such as telephone calls, where delays can raise suspicions. As a result, the Glow-TTS model will take center stage in the following chapters, serving as the primary tool for further testing and analysis.

Chapter 5

Creating a user interface for Coqui TTS

In this chapter, we will discuss some of the problems and limitations that a potential fraudster may encounter with the use of the Coqui TTS program. This chapter will also present a solution to get rid of these limitations to some extent. At the core of the solution is a user interface that allows one to interact with the Coqui TTS program in a more appropriate way with respect to the goals of this thesis.

5.1 Coqui TTS limitations

Coqui TTS provides a wide range of pre-trained models for speech generation and is a console application with the ability to select a model, specify target text, and other parameters. However, the main problem is that the console application works by allowing you to specify input parameters (including target text), loading a pre-trained model, which also takes time, then processing and outputting the generated speech and finishing its work. However, it can be noticed that with this behavior we get quite a big loss of time due to the need to restart the application, input text, parameters, wait for the selected model to be loaded, and finally wait for the processing and output of the result. Looking at this situation through the eyes of the fraudster, it is clear that this loss of time is a big problem for him/her, as a long delay in answering due to the necessary actions described above during a telephone conversation with a „victim“ will definitely arouse suspicion. The question is how can the fraudster try to solve these problems?

5.2 Removing constraints and creating a program

Given the problems that a fraudster might encounter, as defined in the previous subchapter, we can assume that it is necessary to be able to continuously input text that would be passed for processing without the program terminating after each generated block of text.

Thus, it was decided to write a program using models from the Coqui TTS tool, which would allow one to enter text and queue it for the next deepfake to be generated dynamically without blocking input for the duration of the deepfake creation.

For the purposes mentioned above, a simple text field as an interface will be sufficient, where the input text will be written and from where it will be read and sent to the input for a model to generate speech. The basic idea is to break down the whole text into sentences that would be sent to the model as input, one by one, as they are written. To guarantee that different parts of the program do not block each other, it was decided to use the principle of threads, where one thread starts an interface with a text field for data input and writes sentences that are input for the model to the queue, and the other thread processes the text passed and generates a sound signal to the output. The abstract principle of the program is illustrated in Figure 5.1.

In addition to the fact that the program allows continuous text input without the need to restart, the model with which the speech synthesis is performed will also be loaded only once and for the entire duration of the program session with this model.

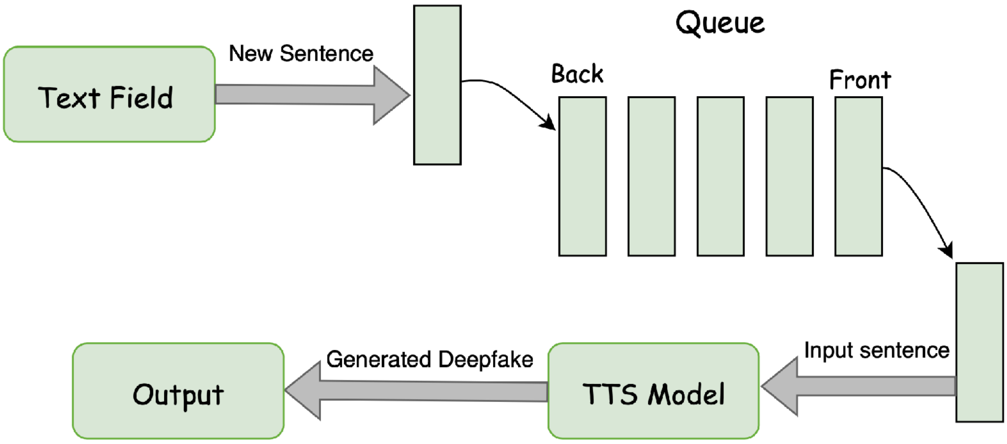
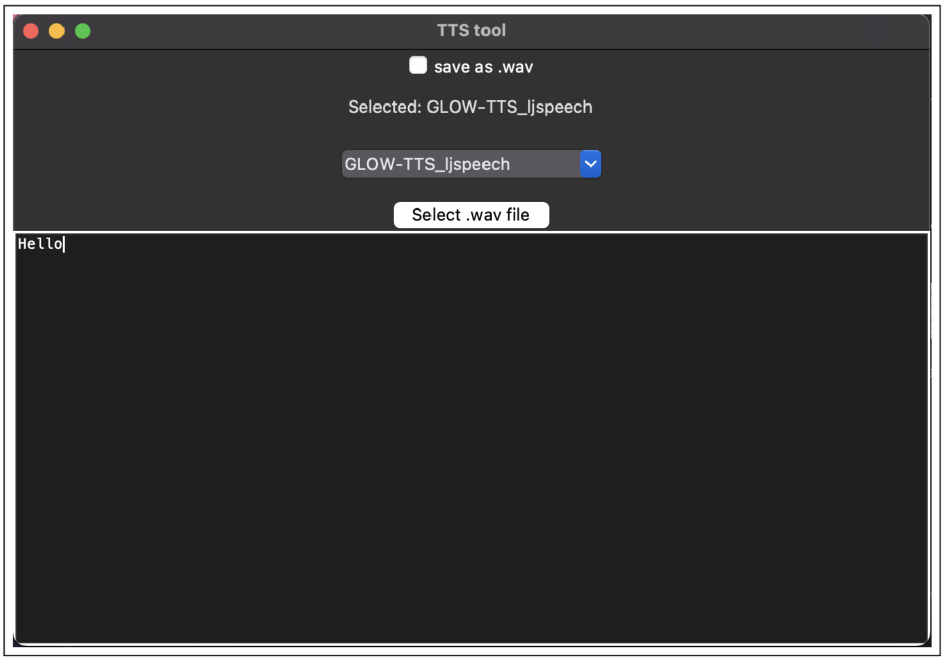


Figure 5.1: Abstract principle of the program operation.

Also, to make it easier to test the program, a functionality with the ability to select different models, each one of them will be loaded without the need to restart the application, has been added. The result of deepfake generating can be output to the computer speakers or saved as a '.wav' file.

Since, based on the experiments in Chapter 4, the best of the tested models in terms of the deepfake generation rate is the Glow-TTS model, the major part of the tests were conducted with it. Within the testing of the program, it was confirmed that the generation rate for the long target texts using the Glow-TTS model is almost real-time and there is no blocking of text input during deepfake generation.

Thus, the created application allows one to generate voice deepfakes using under-the-hood models from the Coqui TTS tool and provides the user the ability to continuously input text for further synthesis. This program was written using the Python programming language, its built-in Tkinter library for creating graphical user interface applications, and of course with the application programming interface (API) of Coqui TTS tool. It can be cautiously concluded that a fraudster with some programming skills can use software that was created for research or entertainment purposes for his own gain. It is also worth to mention that nowadays even programming skills may not be so crucial for writing this kind of program because with the use of large language models(LLMs) like ChatGPT it is possible to achieve significant simplification in writing. The interface of the program is shown in Figure 5.2.



Selected: GLOW-TTS_ljspeech

Hello

Figure 5.2: Graphical user interface of the created program.

5.3 Chapter conclusion

This chapter identified limitations in the usability of Coqui TTS in real-time for applications requiring continuous speech generation. These limitations are related to the work of the console interface and the need to reload the selected model for each text segment. To eliminate these shortcomings, a new program has been developed that takes advantage of the Coqui TTS application programming interface (API) capabilities. This program features a user-friendly interactive interface that simplifies the deepfake creation process.

One of the key improvements is the ability to continuously type text. Users can type sentences without waiting for each deepfake to be generated, which significantly improves workflow efficiency. In addition, the loading of the model has been optimized in the program. The selected model is loaded only once when the program starts, eliminating the need to load it again for each text segment. This optimization, combined with efficient text processing, significantly reduces the time required to generate subsequent deepfakes.

The effectiveness of the program was validated by testing with the Coqui TTS Glow-TTS model. These tests confirmed that the program generates speech in near real time, which means that one of the goals of this thesis has been achieved and that we have succeeded in creating a tool capable of generating near-real-time speech using open-source tools.

Chapter 6

Quality of generated speech

This chapter explores whether a high-quality deepfake can be created using the designed program from Chapter 5. It is necessary to evaluate the quality of the generated speech in two ways: first, we will evaluate how well deepfakes detection methods can determine if the speech from the input is artificial. Second, we will involve human evaluation, real people, to determine if they can detect any signs that the speech is not genuine. By combining these evaluations, we will gain valuable insights into the effectiveness of the used model and identify potential shortcomings of both automatic and human spoof detection.

6.1 Deepfake quality measurement approaches

Returning to the bank example in the introduction 1, where when we call the bank for any purpose related to the manipulation of our personal data or bank account, our voice during the conversation is matched with a sample of our real voice from the bank's database. At the same time, in addition to verification by voice biometrics systems, our conversation partner (bank employee), although unable to accurately compare our voice from the conversation and recorded in the database, can determine whether the speech sounds natural and whether there are no signs in it that could cause suspicion of the employee. Thus, our goal as a fraudster is to try to evaluate the quality of the model used against both voice deepfake detection methods and a real person, i.e. whether the generated speech can fool them both, thus gaining access to personal data or financial transactions of the bank customer whose voice the fraudster may try to imitate.

6.2 Quality assurance through detection methods

To verify the quality of the deepfakes generated by the Glow-TTS model, four pre-trained voice deepfake detection models were selected:

- Resemblyzer [29]
- RDINO [7]
- CAM++ [37]
- ERes2Net [8]

According to the author's findings, these models are the only ones that present the ability to estimate the similarity of audio recordings directly, without the need for model training or dealing with an inappropriate input format.

Resemblyzer from Resemble.ai utilizes advanced machine learning techniques to extract representations of speech signals, allowing robust comparisons between different voices. The tool works by analyzing various acoustic characteristics of the voice, such as pitch, duration, and spectral characteristics. It compares the voice features of each recording with known voice reference samples to determine the likelihood of authenticity of a voice. It uses deep learning algorithms and neural network architectures to detect manipulations or alterations in voice recordings. This makes it a valuable tool in the fight against voice-based misinformation and fraud.

RDINO (Resnet Discriminator with Noise Injection Object) is a deep learning model designed to identify voice deepfakes. RDINO utilizes a Resnet [39] discriminator, a type of convolutional neural network (CNN) known for its effectiveness in image recognition tasks. This Resnet is adapted to analyze audio data. A unique feature of RDINO is the Noise Injection Object. During training, this object injects controlled noise into the audio samples. This helps the model to learn to differentiate between real audio characteristics and inconsistencies introduced by deepfake techniques.

CAM++ model introduces a technique called context-aware masking. This masking approach focuses the network's attention on relevant parts of the speech input while filtering out irrelevant noise. Imagine focusing on the speaker's voice and ignoring background music using a mask. The paper on this model states that context-aware masking helps CAM++ achieve high speaker verification accuracy and be faster and more computationally efficient than other speaker verification systems.

ERes2Net, short for Enhanced Res2Net, is a deep learning architecture designed specifically for speaker verification tasks. It is based on the popular Res2Net [11] architecture, known for capturing multiscale features in data, which is very important for speaker verification when it is important to extract subtle details from speech.

A key innovation of ERes2Net is the mechanism for merging local and global characteristics. This mechanism combines two aspects: local feature fusion (LFF) aims to combine features within a single building block of the network to capture fine details of the voice, while global feature fusion (GFF) combines features from different blocks to learn broader speaker characteristics.

ERes2Net takes this a step further: instead of a simple summarizing or concatenation, it includes an attentional feature fusion module. This module assigns weights to different features based on their importance, potentially leading to more efficient fusion. Overall, ERes2Net aims to achieve better speaker verification performance by combining local and global features more efficiently and using attention mechanisms to focus on relevant speaker information.

To conduct an experiment with the detection methods presented above to determine the quality of the deepfakes created by the Glow-TTS model, it is necessary to create a set of audio recordings that will be divided into recordings of real speech and synthesized speech. The real speech recordings are audio files from the LJ Speech dataset on which the Glow-TTS model was trained. The audio files labeled as deepfake are the speech generated by this model.

The result of the experiment using the Resemblyzer tool is shown in Figure 6.1. Green denotes recordings of a real person's voice, and red denotes synthesized speech. The black dotted line indicates the threshold after which the given audio file is considered to be a real human voice compared to the original. The x-axis shows the audio file names and the y-axis shows the likelihood ratio of each recording with the original voice.

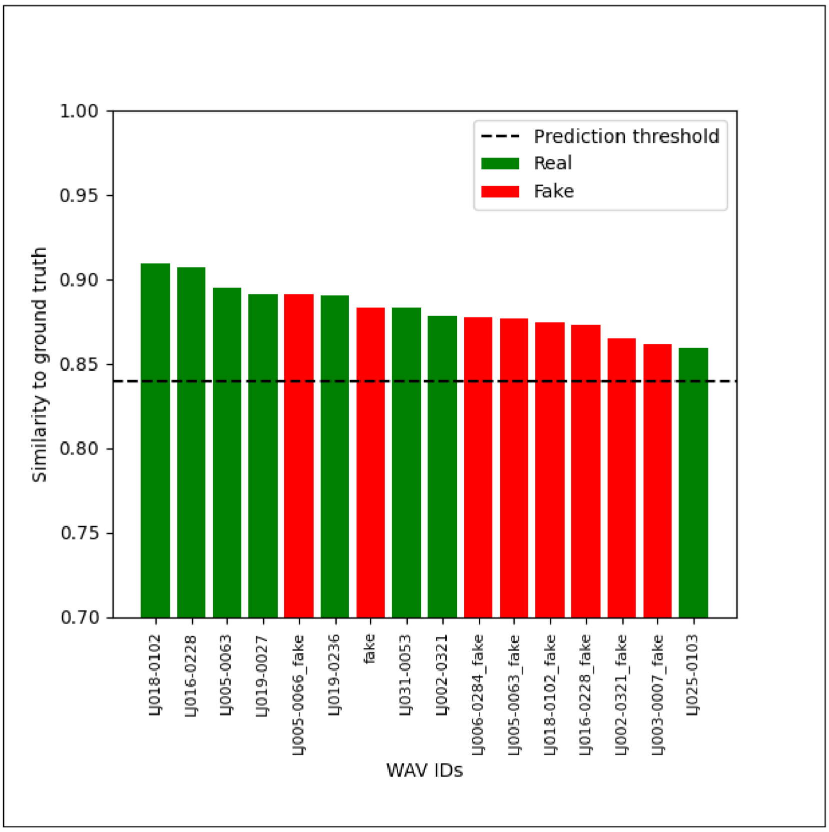


Figure 6.1: Deepfake detection experiment with Resemblyzer.

It can be seen that the Resemblyzer tool was unable to recognize any of the deepfakes presented for the purposes of the experiment as synthesized speech, as for all of them the similarity coefficient was above the threshold value and can be said to be around 86% on average.

Experiments with other deepfake detection models were divided into text-dependent and text-independent tests. Here, the first type of tests contains the original speech and the deepfake with the same target text, and in the other, the target text in the generated speech is different from the text in the original speech. The tests were conducted in pairs in the format: „original - deepfake“, where for each voice deepfake a similarity coefficient is calculated with respect to the original speech from the given pair. For each model, the average value of the similarity coefficient for each of the types of experiments defined above was taken separately. The results of the experiment are shown in Table 6.1.

	similarity score [from 0 to 1]	
model name	**text-dependent**	**text-independent**
RDINO	~ 0.85	~ 0.79
CAM++	~ 0.81	~ 0.79
ERes2Net	~ 0.82	~ 0.79

Table 6.1: Deepfake detection experiments with RDINO, CAM++ and ERes2Net.

From the above results, we can see that all three models for our generated deepfakes for both types of tests (text-dependent or text-independent) give a similarity coefficient around 80%. We can also see that for text-dependent tests, this coefficient is slightly higher than for another test type. It can be concluded that the same phrases uttered by the original and its deepfake affect the result in favor of accepting the deepfake as a real voice.

From the results of the whole experiment, we can conclude that the detection methods used could very likely accept the generated deepfake by the Glow-TTS model as the real voice of a certain person.

6.3 People's ability to identify voice deepfake

Following the promising results obtained in the previous section, where detection models had difficulty distinguishing between real and synthesized speech, we attempted to understand how well people can identify synthetic voices. To measure this, a two-part online survey was designed. Thirteen people participated in the online survey, providing valuable information on human perception of synthesized voices. The abstract visualization of the survey is shown in Figure 6.2.

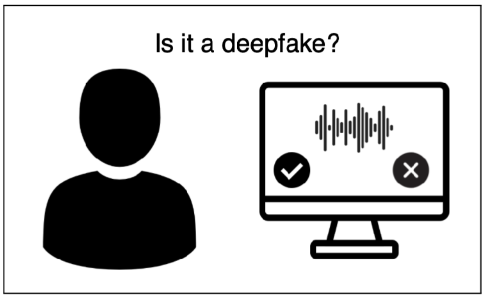


Figure 6.2: Abstract visualization of the survey.

In the first part, the participants listened to ten mixed audio files, half of which contained real speech and the other half generated speech. Their task was to determine which recordings were real and which were synthetic. This initial test provided a basic understanding of a person's ability to recognize voice deepfakes without prior familiarity with the original voice.

In the second part of the survey, it was slightly different. First, participants received a 30-second recording of a real voice. Then, following the same format as in the first part, they listened to ten new audio files. Again, their task was to distinguish the real speech from the synthesized one. In this case, however, the audio files contained recordings of both the original voice and deepfakes reproducing it. This manipulation allowed us to determine how well people can recognize synthetic voices after being exposed to a speaker's natural speech patterns.

After the test, we also included an additional section in which participants could leave brief comments on the quality of the deepfakes. This feedback was intended to identify the specific characteristics of the recordings that raised suspicion and ultimately led them to classify the voice as synthetic. Analyzing these comments along with the test results will provide a more comprehensive picture of an individual's strengths and weaknesses in recognizing voice deepfakes. The results of the survey are shown in Table 6.2.

From the results in the table above, we can see that in the overwhelming number of questions, users were almost 100% likely to identify which of the recordings were real and which were synthesized speech. Based on the comments given by the test takers, the main reasons for determining that the speech was synthetic were: „electric/robotic“ accent, monotone speech, passive emotions in a voice, slow pronunciation, noise, pronunciation glitches. From this we can conclude that deepfakes generated by the Glow-TTS model are unable to deceive people in most cases.

		User answer		
Test part	**Question**	Real	Fake	**Right answer**
first part	1	77 %	23 %	Real
	2	100 %	0 %	Real
	3	0 %	100 %	Fake
	4	0 %	100 %	Fake
	5	0 %	100 %	Fake
	6	100 %	0 %	Real
	7	100 %	0 %	Real
	8	0 %	100 %	Fake
	9	100 %	0 %	Real
	10	0 %	100 %	Fake
second part	1	100 %	0 %	Real
	2	92,3 %	7,7 %	Real
	3	0 %	100 %	Fake
	4	0 %	100 %	Fake
	5	7,7 %	92,3 %	Fake
	6	100 %	0 %	Real
	7	92,3 %	7,7 %	Real
	8	0 %	100 %	Fake
	9	7,7 %	92,3 %	Fake
	10	92,3 %	7,7 %	Real

Table 6.2: Survey results.

6.4 Chapter conclusion

The goal of this chapter was to evaluate the ability of Glow-TTS to generate voice deepfakes that could convincingly mimic real human speech, and thus bypass detection models often used to identify synthetic speech.

Several open-source voice deepfake detection models were used to evaluate the quality of generated speech. The results were quite encouraging: the similarity coefficient between the original and synthesized speech for Glow-TTS exceeded 0.8. Such a high score indicates a significant level of similarity in audio characteristics, suggesting that the model can produce realistic-sounding voice.

However, to gain a more complete picture of perceived quality, we conducted an online survey in which participants were asked to distinguish real audio samples from synthesized ones. This evaluation revealed a crucial aspect that is not considered by detection models: the subjective experience of listeners. Although detection models may have struggled to distinguish between real and synthetic speech based on audio features, human perception proved to be more accurate. Participants identified various defects in some deepfakes created by Glow-TTS, including a lack of natural emotional expression in the voice, slowed pronunciation, and the presence of noise and glitches.

Based on these results, it can be concluded that although Glow-TTS does a good job of deceiving voice recognition models due to its high similarity scores, it fails to produce deepfakes that can convincingly deceive listeners. This indicates that Glow-TTS in its current form is unlikely to be a successful tool for fraudsters seeking to impersonate real people over the phone. However, its ability to bypass detection models highlights the ongoing concern of developing robust methods for detecting more complex deepfakes.

Chapter 7

Discussion of the obtained results

In this chapter, we analyze the results obtained and evaluate their implications on the security of voice biometric authentication systems.

Taking into account the results of the experiments conducted in Chapter 4, we can conclude that although computing power has a significant impact on the rate of speech synthesis, the crucial role is taken by the architecture of the model itself, and only by increasing the computing power it is unlikely to achieve the real-time deepfakes creation for each model. However, it is still a fact that having a relatively high-performance device significantly improves model output time.

It is also worth mentioning that the use of open-source tools provides a lot of opportunities to customize and modify them to suit one's needs, as demonstrated in Chapter 5. Thus, it can be said that a fraudster with some programming skills can use such tools as a basis for creating his own custom tools that can pose a real threat to the security of people's personal data and funds.

Speaking of voice biometrics systems, based on the results of the experiment in Chapter 6, it can be assumed that existing models have a chance of defeating such security tools. However, it is important to mention that the author has not had an opportunity to interact with real voice biometric systems used in real institutions, and therefore does not know how much they outperform publicly available models. It is also worth mentioning that the quality of deepfakes generated by the selected Glow-TTS model can be estimated as average, and some other models are able to generate higher-quality speech, which, however, affects the output time. Knowing these details, we can conclude that since even a model like Glow-TTS was able to fool the voice deepfake detection models used in the experiments with a high chance, the models generating higher quality speech, in the author's opinion, have a serious chance to fool real voice biometric systems as well.

However, the author assumes that the main target of fraudsters will not be institutions, but ordinary people. Since attempting to commit illegal acts against institutions such as banks carries a high risk of being detected, it is much safer to interact with individuals. Based on the results of an experiment with the online survey from Chapter 6 we found that people were 100% likely to be able to recognize which speech was genuine and which was fake. However, it is also worth considering that the quality of the deepfakes generated by the Glow-TTS model, as mentioned above, is average, and for higher-quality deepfakes, the survey result could have been different.

Perhaps the current publicly available programs and models are not capable of generating voice deepfakes in real time and with high enough quality at the same time to be able to fool both voice biometrics systems and humans, but the author supposes that it is only a matter of time before tools that are capable of doing so become publicly available, probably even in the near future given the rapid advances in deep learning. Thus, organizations dealing with extremely sensitive personal data, such as banks, as well as ordinary people should definitely prepare for the potential surge in phone fraud and the use of people's voices for illegal purposes.

Chapter 8

Conclusion

This thesis presented the problem of the growing quality of voice deepfake models and their availability on the Internet from the point of view of their use for illegal purposes. When studying the question of how easy it is to generate real-time speech using open access tools and to deceive voice biometrics systems and humans, the author of this paper decided to put himself in the position of a fraudster.

The main technologies for generating artificial speech were identified: text-to-speech (TTS) and voice conversion (VC). In addition, open-source tools capable of generating voice deepfakes using the above technologies were selected. On the basis of these tools and models they contain, it was decided to test the hypothesis that the time of deepfake creation depends on the computing power of the computer. According to the results of the experiments, this hypothesis was confirmed and a model capable of generating speech almost in real time was found.

Since the software tool containing this model had some disadvantages for continuous generation of synthesized speech in real time, it was decided to write our own program, which would be a convenient interface and add-on over the original program, eliminating the disadvantages of the original program in terms of generating deepfakes in real time. The created program allows continuous input of the target text, which is alternately fed to the input of the model for subsequent generation of voice deepfakes.

Next, it was necessary to evaluate the quality of the speech synthesized by the selected model. The author decided to test both detection methods and people's ability to recognize deepfake. Tests conducted on synthetic speech detection methods showed results in favor of our model, however, according to the results of the created online survey, most of the people were able to distinguish between the speech of a real person and an artificial one.

Finally, the author evaluated the results achieved and, based on them, put forward his assumptions about the risks and threats that the kind of tools discussed in this paper pose to the security of people's personal data and funds.

Bibliography

[1] Anderson, M. Deepfaked Voice Enabled $35 Million Bank Heist in 2020. *UNITE.AI* [online]. 15. october 2021 [cit. 2023-11-20]. Available at: https://www.unite.ai/deepfaked-voice-enabled-35-million-bank-heist-in-2020/.

[2] Arik, S., Diamos, G., Gibiansky, A., Miller, J., Peng, K. et al. *Deep Voice 2: Multi-Speaker Neural Text-to-Speech* [online]. 2017 [cit. 2024-01-26]. Available at: https://arxiv.org/abs/1705.08947.

[3] Bhatt, S., Jain, A. and Dev, A. Acoustic Modeling in Speech Recognition: A Systematic Review. *International Journal of Advanced Computer Science and Applications* [online]. The Science and Information Organization. november 2020, vol. 11, no. 4, p. 397–412, [cit. 2023-11-15]. DOI: 10.14569/IJACSA.2020.0110455. Available at: http://dx.doi.org/10.14569/IJACSA.2020.0110455.

[4] Bilcu, E. B. *Text-To-Phoneme Mapping Using Neural Networks* [online]. Tampere, 2008. [cit. 2023-11-15]. Dissertation. Tampere University of Technology. ISBN 978-952-15-2045-7. Available at: https://trepo.tuni.fi//handle/10024/114031.

[5] Casanova, E., Weber, J., Shulby, C., Junior, A. C., Gölge, E. et al. *YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for everyone* [online]. 2023 [cit. 2023-11-30]. Available at: https://arxiv.org/abs/2112.02418.

[6] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S. et al. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing* [online]. Institute of Electrical and Electronics Engineers (IEEE). october 2022, vol. 16, no. 6, p. 1505–1518, [cit. 2023-12-03]. DOI: 10.1109/jstsp.2022.3188113. ISSN 1941-0484. Available at: http://dx.doi.org/10.1109/JSTSP.2022.3188113.

[7] Chen, Y., Zheng, S., Wang, H., Cheng, L. and Chen, Q. Pushing the limits of self-supervised speaker verification using regularized distillation framework. In: IEEE. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* [online]. 2023, p. 1–5 [cit. 2024-02-20]. Available at: https://arxiv.org/abs/2211.04168.

[8] Chen, Y., Zheng, S., Wang, H., Cheng, L., Chen, Q. et al. An Enhanced Res2Net with Local and Global Feature Fusion for Speaker Verification. [online]. 2023, [cit. 2024-02-22]. Available at: https://arxiv.org/abs/2305.12838.

[9] Chung, J. S., Nagrani, A. and Zisserman, A. VoxCeleb2: Deep Speaker Recognition. In: *INTERSPEECH* [online]. 2018 [cit. 2024-04-01]. Available at: https://www.robots.ox.ac.uk/~vgg/publications/2018/Chung18a/chung18a.pdf.

[10] Coqui, T. *Coqui TTS* [online]. 2023 [cit. 2024-03-21]. Available at: https://github.com/coqui-ai/TTS.

[11] Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H. et al. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. Institute of Electrical and Electronics Engineers (IEEE). 2021, vol. 43, no. 2, p. 652–662, [cit. 2024-02-25]. DOI: 10.1109/tpami.2019.2938758. ISSN 1939-3539. Available at: http://dx.doi.org/10.1109/TPAMI.2019.2938758.

[12] Ito, K. and Johnson, L. *The LJ Speech Dataset* [online]. 2017 [cit. 2024-04-04]. Available at: https://keithito.com/LJ-Speech-Dataset/.

[13] Jemine, C. *Real-Time Voice Cloning* [online]. Liège, Belgium, 2019. [cit. 2023-11-26]. MASTER THESIS. Université de Liège. Available at: https://matheo.uliege.be/handle/2268.2/6801?locale=en.

[14] Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J. et al. *Transfer learning from speaker verification to multispeaker text-to-speech synthesis* [online]. 2018 [cit. 2023-11-25]. Available at: https://proceedings.neurips.cc/paper_files/paper/2018/file/6832a7b24bc06775d02b7406880b93fc-Paper.pdf.

[15] Kim, J., Kim, S., Kong, J. and Yoon, S. *Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search* [online]. 2020 [cit. 2023-11-28]. Available at: https://arxiv.org/abs/2005.11129.

[16] Kim, J., Kong, J. and Son, J. *Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech* [online]. 2021 [cit. 2023-11-28]. Available at: https://arxiv.org/abs/2106.06103.

[17] Kong, J., Kim, J. and Bae, J. *HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis* [online]. 2020 [cit. 2023-11-28]. Available at: https://arxiv.org/abs/2010.05646.

[18] Lehiste, I. and Lass, N. J. Suprasegmental features of speech. In: Lass, N. J., ed. *Contemporary Issues in Experimental Phonetics* [online]. Columbus: Academic Press, 1976, vol. 225, p. 225–239 [cit. 2023-11-16]. DOI: https://doi.org/10.1016/B978-0-12-437150-7.50013-0. ISBN 978-0-12-437150-7. Available at: https://books.google.cz/books?hl=cs&lr=&id=DYT7GSNMdhAC&oi=fnd&pg=PA225&dq=suprasegmental+features+speech&ots=m9IEfpm-D-&sig=FqFNCcfkIaa_vFr9AVq1fVF1vQI&redir_esc=y#v=onepage&q=suprasegmental%20features%20speech&f=false.

[19] li, J., tu, W. and xiao, L. *FreeVC: Towards High-Quality Text-Free One-Shot Voice Conversion* [online]. 2022 [cit. 2023-12-03]. Available at: https://arxiv.org/abs/2210.15418.

[20] Nagrani, A., Chung, J. S. and Zisserman, A. VoxCeleb: a large-scale speaker identification dataset. In: *INTERSPEECH* [online]. 2017 [cit. 2024-04-01]. Available at: https://www.robots.ox.ac.uk/~vgg/publications/2017/Nagrani17/nagrani17.pdf.

[21] Numediart. *MBROLA* [online]. 2019 [cit. 2024-03-21]. Available at: https://github.com/numediart/MBROLA.

[22] Oord, A. van den, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O. et al. WaveNet: A Generative Model for Raw Audio. New York: Cornell University. september 2016, [cit. 2023-12-20]. Available at: https://arxiv.org/pdf/1609.03499v2.pdf.

[23] Panayotov, V., Chen, G., Povey, D. and Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* [online]. 2015, p. 5206–5210 [cit. 2024-04-01]. DOI: 10.1109/ICASSP.2015.7178964. Available at: https://ieeexplore.ieee.org/abstract/document/7178964.

[24] Pierrehumbert, J. Phonological and phonetic representation. *Journal of Phonetics* [online]. 1st ed., version 1.0. Evanston: Cambridge University Press. july 1990, vol. 18, no. 3, p. 375–394, [cit. 2023-11-14]. Journal of Phonetics, no. 1. DOI: https://doi.org/10.1016/S0095-4470(19)30380-8. ISSN 0095-4470. Phonetic Representation. Available at: https://www.sciencedirect.com/science/article/pii/S0095447019303808.

[25] Play.ht, T. Generate AI Voices, Indistinguishable from Humans. *Play.ht* [online]. [cit. 2024-03-20]. Available at: https://play.ht.

[26] Prenger, R., Valle, R. and Catanzaro, B. Waveglow: A Flow-based Generative Network for Speech Synthesis. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* [online]. Brighton, UK: IEEE, 2019, p. 3617–3621, 17.04.2019 [cit. 2023-11-26]. DOI: 10.1109/ICASSP.2019.8683143. ISBN 978-1-4799-8131-1. Available at: https://ieeexplore.ieee.org/abstract/document/8683143.

[27] Ramu Reddy, V. and Sreenivasa Rao, K. Prosody modeling for syllable based text-to-speech synthesis using feedforward neural networks. *Neurocomputing* [online]. Kharagpur 721302, West Bengal, India: Indian Institute of Technology Kharagpur. january 2016, vol. 171, p. 1323–1334, [cit. 2023-11-10]. DOI: https://doi.org/10.1016/j.neucom.2015.07.053. ISSN 0925-2312. Available at: https://www.sciencedirect.com/science/article/pii/S0925231215010395.

[28] Resemble, T. Generative Voice AI built for Enterprise. *RESEMBLE.AI* [online]. [cit. 2024-03-20]. Available at: https://www.resemble.ai.

[29] Resemble, T. *Resemblyzer* [online]. 2019 [cit. 2024-02-20]. Available at: https://github.com/resemble-ai/Resemblyzer/tree/master.

[30] Respeecher, T. AI voices that have impact. *Respeecher* [online]. [cit. 2024-03-28]. Available at: https://www.respeecher.com.

[31] Sasirekha, D. and Chandra, E. Text to speech: a simple tutorial. *International Journal of Soft Computing and Engineering (IJSCE)* [online]. Citeseer. march 2012, vol. 2, no. 1, p. 275–278, [cit. 2023-11-08]. ISSN 2231-2307. Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e7ad2a63458653ac965fe349fe375eb8e2b70b02.

[32] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N. et al. *Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions* [online]. 2018 [cit. 2023-11-24]. Available at: https://arxiv.org/abs/1712.05884.

[33] Sisman, B., Yamagishi, J., King, S. and Li, H. An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* [online]. IEEE. 2021, vol. 29, p. 132–157, [cit. 2023-11-17]. DOI: 10.1109/TASLP.2020.3038524. Available at: https://ieeexplore.ieee.org/abstract/document/9262021.

[34] Smith, J. O. *Physical Audio Signal Processing* [online]. 2010 editionth ed. W3K Publishing, 2010 [cit. 2023-11-17]. Dudley's Vocoder. ISBN 978-0-9745607-2-4. Online book, 2010 edition. Available at: https://ccrma.stanford.edu/~jos/pasp/.

[35] Vassil Panayotov, D. P. *LibriSpeech ASR corpus* [online]. 2015 [cit. 2024-04-01]. Available at: https://www.openslr.org/12.

[36] Veaux, C., Yamagishi, J., MacDonald, K. et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. [online]. University of Edinburgh. The Centre for Speech Technology Research (CSTR). 2017, [cit. 2024-04-04]. Available at: https://datashare.ed.ac.uk/handle/10283/2651.

[37] Wang, H., Zheng, S., Chen, Y., Cheng, L. and Chen, Q. CAM++: A Fast and Efficient Network for Speaker Verification Using Context-Aware Masking. [online]. 2023, [cit. 2024-02-21]. Available at: https://arxiv.org/abs/2303.00332.

[38] Wang, Y., Skerry Ryan, R., Stanton, D., Wu, Y., Weiss, R. J. et al. Tacotron: Towards End-to-End Speech Synthesis. New York: Cornell University. march 2017, [cit. 2023-12-20]. Available at: https://arxiv.org/pdf/1703.10135.pdf.

[39] Zhu, H., Sun, M., Fu, H., Du, N. and Zhang, J. Training a Seismogram Discriminator Based on ResNet. *IEEE Transactions on Geoscience and Remote Sensing* [online]. 2021, vol. 59, no. 8, p. 7076–7085, [cit. 2024-02-24]. DOI: 10.1109/TGRS.2020.3030324. Available at: https://ieeexplore.ieee.org/abstract/document/9246206.