SPEAKER ANONYMIZATION USING GENERATIVE ADVERSARIAL NETWORKS

BY

AYA ALJA'FARI

SUPERVISOR

DR. AMJED AL-MOUSA

CO-SUPERVISOR

PROF. IYAD JAFAR

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Princess Sumaya University for Technology

King Abdullah I School of Graduate Studies and Scientific Research

June 2021

Authorization Form

I , **Aya ALJa'fari**, authorize Princess Sumaya University for Technology to supply copies of my MSc. thesis to libraries, establishments or individuals on request, according to the Regulations of Princess Sumaya University for Technology.

Signature: 

Date: 2.10.2021

Committee Decision

This Thesis **"Speaker Anonymization Using Generative Adversarial Networks"** was Successfully Defended and Approved on **June 3, 2021**.

Committee Members	**Signature**
Dr. Amjed Al-Mousa, Supervisor Associate Professor of Computer Engineering	Amjed Al-mousa .....
Prof. Iyad Jafar, Co-Supervisor Professor of Computer Engineering The University of Jordan	iyadjafar .....
Prof. Gheith Abandah, External Examiner Professor of Computer Engineering The University of Jordan	G Abandah .....
Prof. Ali Alhaj, Internal Examiner Professor of Computer Engineering	Ali Al-Haj .....
Dr. Awos Kanan, Internal Examiner Assistant Professor of Computer Engineering	A. Kanan .....

Dedication

This is for you, Mom and Dad.
Thank you for always believing in me.

Acknowledgments

During the journey of my thesis, I was fortunate to have the support and guidance of not one but two professors. I would like to first begin by thanking my supervisors Dr. Amjed Al-Mousa and professor Iyad Jafar for their priceless insights, patience, and trust towards every aspect of my thesis and research work.

I would like to express my sincere gratitude to my family. Had it not been for your support, understanding, and unwavering faith in my potential, I would not have gotten this far.

I would also like to acknowledge my colleagues at Samsung R&D Institute, Shorouq Sabbah, Mohammed Hamdan, Osama Sabri, Ali AlMasaeed, Anas Toma, and Mohammed Darwish. Had it not been for the great environment at Samsung, combining a master's degree with a full-time job would not have been possible. Working with you is both a privilege and a pleasure. Thank you for your continuous support and understanding during my master's journey.

Last but not least and by virtue of my thesis work, I've had the pleasure of interacting with the organizers of the VoicePrivacy Challenge 2020. You have spared no expense in answering my questions and your feedback was invaluable and for that, you have my heartfelt thanks.

Aya AlJa'fari

List of Tables

Table 2.1: Network Architecture of the x-vector System 15

Table 3.1: The Four GAN Designs' Architectures and Hyperparameters 29

Table 4.2: Statistics of the Training Datasets of the x-vector Anonymization Model. . . . 35
Table 4.1: Anonymization System: Model Description and Training Corpora. 35
Table 4.3: Statistics of the Validation Datasets [5]. 37
Table 4.4: Statistics of the Test Datasets [5]. 37
Table 4.5: GAN Generated Pool: ASR scores 41

Table A.1: GAN-generated Pool: ASV Results for Development Data. 58
Table A.2: ASV Results for CTGAN-generated Pool. 59
Table A.3: ASR Results for CTGAN-generated Pool with Diffrent Pool Sizes. 60
Table A.4: ASV Results for PLDA and Cosine Distances 61
Table A.5: ASR Results for PLDA and Cosine Distances 62
Table A.6: ASV Comparison for CTGAN against GMM and Baseline 62
Table A.7: ASR Comparison for CTGAN against GMM and Baseline 62

List of Figures

Figure 2.1: Architecture of a Smart Assistant 6
Figure 2.2: Basic Operation of the ASR System 8
Figure 2.3: Architecture of the ASR Engine 9
Figure 2.4: Speech Signal Processing Functionalities 12
Figure 2.5: Basic Speaker Verification System 13
Figure 2.6: Equal Error Rate (EER) 16
Figure 2.7: Speaker Anonymization Pipeline [4] 20

Figure 3.1: Generative Adversarial Network Architecture 23
Figure 3.2: Proposed Anonymization System Design 27

Figure 4.1: KS Score Results for Different GAN Designs 38
Figure 4.2: Cross-Cosine Similarity Distribution for Four GAN Designs 39
Figure 4.3: GAN Generated Pool: EER scores 41
Figure 4.4: EER Scores for CTGAN Generated Pool on Development Sets 42
Figure 4.5: EER Scores for CTGAN Generated Pool on Test Sets 43
Figure 4.6: ASR results for CTGAN Generated Pool with Different Pool sizes 44
Figure 4.7: ASV Results for PLDA and Cosine Distances on Development Sets 45
Figure 4.8: ASV Results for PLDA and Cosine Distances on Test Sets 46
Figure 4.9: ASR Results for PLDA and Cosine Distances 47
Figure 4.10: ASV Comparison for CTGAN against GMM and Baseline 48
Figure 4.11: ASR Comparison for CTGAN against GMM and Baseline 48

List of Appendices

Appendix A: Tabulated Results 58

List of Abbreviations

ANN Artificial Neural Network.

AR Autoregressive.

ASR Automatic Speech Recognition.

ASV Automatic Speaker Verification.

BN Bottleneck.

CGAN Conditional Generative Adversarial Network.

CLLR Log-Likelihood-Ratio Cost Function.

CNN Convolutional Neural Network.

CTGAN Conditional Tabular GAN.

DARPA Defense Advanced Research Project Agency.

DM Dialog Manager.

DNN Deep Neural Networks.

DWT Discrete Wavelet Transform.

EEA European Economic Area.

EER Equal Error Rate.

EU European Union.

F0 Fundamental Frequency.

FAR False Acceptance Ratio.

FRR False Rejection Ratio.

GAN Generative Adversarial Network.

GDPR General Data Protection Regulation.

GMM Gaussian Mixture Models.

HMM Hidden Markov Model.

KS Kolmogorov-Smirnov.

LLR Log-Likelihood Ratio.

LPC Linear Predictive Coding.

LSTM Long-Short Term Memory.

MFCC Mel Frequency Cepstral Coefficients.

MRFR Market Research Future.

NLU Natural Language Understanding.

PCA Principal Component Analysis.

PDF Probability Density Function.

PPG Phoneme Posteriogram.

RNN Recurrent Neural Network.

RTF Real-Time Factor.

SITW Speakers In The Wild.

SPA Smart Personal Assistants.

SRE Speaker Recognition Evaluation.

SVM Support Vector Machines.

TFD Toronto Face Data Set.

TTS Text-To-Speech.

UBM Universal Background Model.

VQ Vector Quantization.

VT Voice Transformation.

WER Word Error Rate.

WGAN Wasserstein Generative Adversarial Network.

WRR Word Recognition Rate.

List of Contents

Authorization Form iii

Committee Decision iv

Dedication v

Acknowledgments vi

List of Tables vii

List of Figures viii

List of Appendices ix

List of Abbreviations x

List of Contents xiii

Abstract xvi

Chapter 1: Introduction 1

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Contribution 2

1.4 Organization 3

Chapter 2: Background and Literature Review 4

2.1 Background 4

2.1.1 Smart Assistants 4

2.1.1.1 Architecture . . . 5
2.1.1.2 Market Trend . . . 6
2.1.2 Automatic Speech Recognition . . . 6
2.1.2.1 History . . . 7
2.1.2.2 Mathematical Model . . . 7
2.1.2.3 ASR Evaluation Metrics . . . 10
2.1.2.4 Benchmark Datasets . . . 11
2.1.3 Automatic Speaker Recognition . . . 11
2.1.3.1 Automatic Speaker Verification . . . 13
2.1.3.2 Speaker Identity . . . 14
2.1.3.3 ASV Evaluation Metrics . . . 15
2.1.3.4 ASV Benchmark Dataset . . . 17
2.2 Literature Review on Speech Privacy Preservation . . . 17
2.2.1 Speech Privacy Techniques . . . 17
2.2.2 The VoicePrivacy Challenge . . . 19

Chapter 3: The Proposed Speaker Anonymization Model . . . 22
3.1 Generative Adversarial Networks . . . 22
3.2 Anonymization Component Design . . . 26
3.2.1 System Architecture . . . 26
3.2.2 GAN Selection . . . 28

Chapter 4: Results and Discussion . . . 30
4.1 Experimental Setup . . . 30
4.2 Evaluation Methods . . . 30
4.2.1 Generative Model Evaluation . . . 31
4.2.2 Speaker Anonymization Evaluation . . . 32
4.3 Data Preprocessing . . . 34
4.3.1 Training Data . . . 34
4.3.2 Development Data . . . 36
4.3.3 Evaluation Data . . . 37
4.4 Experimental Results . . . 37
4.4.1 Generative Model Evaluation Results . . . 38
4.4.2 Speaker Anonymization Evaluation Results . . . 40
4.4.2.1 GAN Pool Evaluation . . . 40
4.4.2.2 CTGAN Pool Evaluation . . . 41
4.4.3 Comparison with Previous Work . . . 47

Chapter 5: Conclusion and Future Work 49
5.1 Conclusion 49
5.2 Future Work 50

Bibliography 52

Appendices 58

Speaker Anonymization Using Generative Adversarial Networks

By
Aya AlJa'fari

Supervisor
Amjed Al-Mousa
Co-Supervisor
Iyad Jafar

Abstract

The advent use of smart devices has enabled the production of a wealth of applications that facilitate user interaction in various forms. Speech, the most natural and common form of interaction, reveals private and sensitive information about the user, therefore leaking poses a risk on the user's freedom of speech. Speech may be acquired and used with speech synthesis systems to produce speech recordings that reflect the same user's speaker identity and can be used to attack speaker verification systems. One solution is to anonymize the speaker by hiding his identity from speech before sharing it. In this thesis, synthesized fake identities with audible human voices are used to anonymize speech. The proposed method relied on using adversarial training to optimize the production of fake identities that would enhance the anonymization process. These fake identities are generated using a Generative Adversarial Network (GAN).

Several GAN types were investigated for this purpose and the conditional tabular GAN (CTGAN) showed the best performance among all GAN types according to different metrics. Experimental results proved the ability of the proposed anonymization approach to outperform the best available anonymization systems in terms of the ability to produce a diverse amount of speaker identities (cross cosine similarity distribution, average of 0.75), the closeness between the fake and real identities PDFs (0.55,0.42 KS score for female, males) and the word-error-rate assessed by an external ASR system, achieving 6.27% and 6.5% on libri-dev and libri-test benchmarks, respectively.

Keywords: Speaker anonymization, Voice Privacy, Generative Adversarial Networks, CTGAN, X-vector.

Chapter 1

Introduction

1.1 Motivation

Many technological interfaces nowadays support voice-driven interactions. Speech recordings are being collected from smartphones, televisions, watches, and smart cars. This suggests that speech recordings are transmitted over networks and stored in servers or processed by third-party cloud-based infrastructure [1]. Moreover, these technologies support seamless user interaction where the user can start speaking without pushing any button, which means that these devices are always listening and processing data to detect the presence of a wake-up word. Therefore, there is a great risk for speech data to be exposed.

Systems that understand speech and transcribe it into textual format are referred to as speech recognition systems, whereas systems that identify the identity of the speaker are called speaker recognition systems. Speaker recognition technology is becoming increasingly ubiquitous [2], being used for authenticating individuals and access control across a broad range of different services and devices, e.g., telephone banking services and smart devices that either contain or provide access to personal or sensitive data.

Despite the clear advantages and spread of biometrics technology, persisting concerns regarding intrusions into privacy have dented public confidence. Intrusions into personal privacy are clearly unacceptable and the responsibility to preserve privacy is now demanded in the recent EU General Data Protection Regulation (European Parliament and Council, 2016a, GDPR) [3]. The GDPR is the regulation in the EU on data protection and privacy in the European Union (EU) and the European Economic Area (EEA) that addresses the transfer of personal data outside the EU and EEA areas. It aims primarily to give control to individuals over their personal data. Adequate privacy preservation is therefore essential to ensure that sensitive bio-metric data, including voice recordings

or speech data, are properly protected from misuse. However, privacy preservation schemes of speech data shouldn't compromise the seamless experience of such voice-driven systems.

Speaker anonymization systems are systems that provide a form of speech privacy preservation as they suppress the original identity of the speaker by making the utterances sound as if they were uttered by someone else while leaving all other information in a speech signal intact. The baseline for such technology was first provided by The VoicePrivacy INTERSPEECH 2020 challenge [4], in an attempt to gather effort in the research community to facilitate the development of voice privacy preservation technology, in a response to the recent EU legislation on the protection of personal data.

1.2 Problem Statement

Speaker anonymization typically aims at suppressing the speaker's identity (timbre, pitch, speaking rate, and speaking style) by modifying the original speech signal to make it sound like an anonymous speaker while maintaining the linguistic content, speech quality, and naturalness. Moreover, anonymization should be robust against attackers who try to determine the original speaker's identity. The privacy preservation task is typically formulated as a game involving one or more users who publish some data and an attacker (also called an adversary) who gains legal or illegal access to this data or to derived data and attempts to infer personal information about the users [5].

The VoicePrivacy challenge 2020 addressed the anonymization problem and provided a baseline system for concealing speaker identity by disentangling the linguistic content from the speaker identifiable information. They replaced the speaker identity i.e. x-vector, which is a neural network-based latent representation, with another speaker's identity called a pseudo-identity. Their work was based on the use of a pool of real speakers' identities in which anonymization took place using some distancing techniques. The problem with their approach is the use of real speaker identities in the anonymization process, in which it can be argued that anonymized speech can protect the identity of one user at the expense of exposing another identity. Therefore, we propose another level of anonymity introduced by a generative adversarial network that's trained to synthesize fake human audible x-vectors.

1.3 Contribution

The main contribution in this thesis is that we managed to replace the use of static identities with a method to generate synthetic fake identities with audible human voice characteristics that improved

the performance of speaker anonymization. We believe that this is the first attempt in synthesizing human speaker identities using generative adversarial networks (GAN).

Several GAN types were investigated for this purpose and the conditional tabular GAN (CTGAN) showed the best performance among all GAN types according to different metrics. Experimental results proved the ability of the proposed anonymization approach to outperform the best available anonymization systems in terms of the ability to produce a diverse amount of speaker identities (cross cosine similarity distribution, average of 0.75), the closeness between the fake and real identities PDFs (0.55,0.42 KS score for female, males) and the word-error-rate assessed by an external ASR system, achieving 6.27% and 6.5% on libri-dev and libri-test benchmarks, respectively.

1.4 Organization

The rest of the thesis is organized as follows. In Chapter 2, a general background of the speaker anonymization problem is presented in addition to providing a survey of the related literature. Chapter 3 discusses different generative adversarial networks (GAN) designs and the methodology followed towards developing an effective speaker identity generation model. Chapter 4 outlines several metrics that are used to evaluate different GAN designs followed by a discussion of the findings of the conducted experiments. Chapter 5 summarizes the main findings of this thesis. Finally, and discusses some of the possible future directions of the work.

Chapter 2

Background and Literature Review

This chapter reviews the background and literature relevant to the speaker anonymization problem. The first section provides a background on smart assistants. The section begins with a discussion of the development history of smart assistants' industry and a general overview of its underlying sub-blocks. The second section draws a canvas of the available literature in the area of speaker anonymization. An outline of the work done in this thesis in the context of the literature is also provided.

2.1 Background

In this section, we discuss the essential topics that define the background for the speaker anonymization problem. First, we introduce smart assistants as a pipeline and as a business product. Then, we take a closer look at the speech signal and define the information we can extract from it, namely, speech recognition and speaker recognition. We discuss design details and different use cases for each module.

2.1.1 Smart Assistants

In the past decade, digitization has revolutionized the world around us by bringing new possibilities that ease our daily life activities. Modern software products that understand natural language voice commands and complete tasks for the user are called Smart Personal Assistants (SPA) [6] or Virtual Assistants. They rely on emerging technologies such as artificial intelligence, natural language processing, and speech recognition. A typical virtual assistant is able to interpret human speech, deduce the intent, perform an action and respond via synthesized speech. Actions might include

asking questions, controlling home devices, or managing tasks such as writing an email, playing a song, calling someone, or setting a reminder. Popular virtual assistants nowadays are Google's Assistant, Amazon's Alexa, Apple's Siri, Microsoft's Cortana, and Samsung's Bixby.

Virtual assistants could be deployed on the cloud or on devices. A cloud-based service typically requires an internet connection to work, as the user's commands get transferred to the cloud via the network, processed on the server, and then transferred back to the device through the network. In contrast, the on-device service executes the whole process without the need for an internet connection as the service is installed on the device's hardware. The deployment choice usually demands architectural changes. For instance, designing an on-device model requires careful consideration of latency, memory footprint, and processing power. Such design limitations should be taken into account while developing an on-device service that's at least as accurate as the cloud-based one. On-device deployment choice has a number of advantages such as personalizing the service according to the user's data and preferences (contact list, most used applications, user's speech style, and accent, etc.) and preserving the user's privacy as the data won't travel through the network or get processed on the cloud.

A virtual assistant could be a service on a mobile device or could be hosted on a dedicated device providing this service, just like Amazon's Echo Virtual Assistant. The typical user experience scenario would involve the user calling out the wake word, "Okay Google" for Google's Assistant, for example, then the wake-up model will trigger the launch of the smart assistant, ready to receive a command which typically involves a simple language request. In the case of a cloud-based service, this command is processed and stored in the cloud. Service providers usually keep track of the data and try to make use of it to enhance their systems. In developing AI-based services, the most challenging part of the process is to mitigate differences between training data and testing data (usually a subset from a real scenario). The best scenario would be if the system can train on real-world data i.e. usage data.

2.1.1.1 Architecture

Figure 2.1 depicts the basic architecture of the virtual assistant. Basically, the pipeline consists of four main blocks: Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Dialog Manager (DM), and Text-to-Speech (TTS).

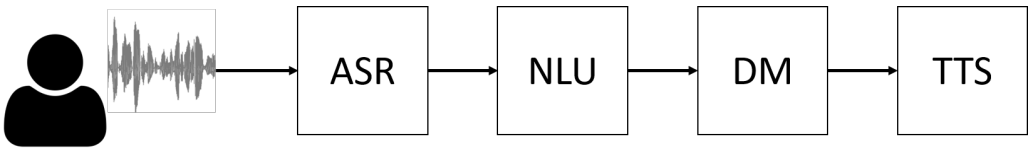

Figure 2.1: Architecture of a Smart Assistant

First, the spoken audio is converted to textual representation via the ASR Engine. Then, the text is "parsed" to understand the request ("intent") and label any data ("entities") by an NLU engine. The DM is responsible for the reasoning. For example, it should answer questions like: Is the user's request complete? Does the virtual assistant need to ask further questions? Can processing the request proceed? What is the needed response? The dialog manager is also responsible for incorporating context in reasoning such as considering the presence of contextual elements (time and location). Finally, the TTS engine is needed in case the response of the virtual assistant is of spoken nature.

2.1.1.2 Market Trend

Studying patterns in the market aids in realizing the importance of investing in current technologies and helps forecast upcoming challenges. Google says that 27% of the global online population is using voice search on mobile [7]. In 2020, STATISTA found that approximately 64% of surveyed experts within the industries of E-learning and market research used speech-to-text automated transcription [8] whereas TECHCRUNCH reports that the number of devices with voice assistants installed in 2020 was over two billion [9].

According to Market Research Future (MRFR) [10], the size of the global speech recognition market is expected to reach $16 billion by 2023. In 2018, the global Captioning and Subtitling Solutions market size was 220 million USD and it is expected to reach 370 million USD by the end of 2025, according to Valuates Reports [11]. The Microsoft market intelligence team found that 41% of users report concerns around trust, privacy, and passive listening [12].

2.1.2 Automatic Speech Recognition

In this section, we provide an overview of the history, mathematical model, evaluation metrics, and benchmark datasets related to automatic speech recognition.

2.1.2.1 History

Since the 1950s, computer scientists have been researching ways to make computers understand human speech. The initial efforts for building an ASR go back to the 1950s where attempts ranged from recognizing isolated digits from a single speaker at Bell Labs [13], recognizing 10 different syllables of a single speaker relying on spectral measurements at RCA Labs [14], building a phoneme recognizer to recognize four vowels and nine consonants using a spectrum analyzer and a pattern matcher at University College in England [15] and the development of a phoneme recognizer in MIT that recognizes ten phonemes in a speaker-dependent manner [16]. In the 1960s, researchers at RCA labs proposed a solution to the non-uniformity of time scales in speech, where they proposed a set of elementary methods to detect speech starts and ends [17]. In the late 1960s, Vintsyuk attempted to solve the problem of aligning two speech utterances using dynamic programming [18].

In the 1970s, several milestones were achieved. The Russians have made the discrete utterance recognition system feasible and stable to use and also helped advance the use of pattern recognition ideas in speech recognition[19]. AT&T Bell Labs researchers used some complex clustering algorithms to determine the number of unique patterns required to identify all variations of different words across a language in an attempt to create speech recognition systems that are truly speaker-independent [20].

In the 1980s, a change in technology occurred by switching from time wrapping template-based techniques to statistical modeling methods; especially the Hidden Markov Model (HMM). The Defense Advanced Research Project Agency (DARPA) community has led a large research project in the 1990s to enhance continuous speech recognition systems and shifted their emphasis to developing natural language front ends to the recognition system. At about the same time, speech technology has been increasingly used within telephone networks to automate and enhance the operators' service [21].

In the early years of the 2000s, the HMM complemented with a feed-forward artificial neural network (ANN) has been the adopted architecture for the state-of-the-art ASR [22]. Nowadays, long-short term memory (LSTM), a type of recurrent neural network (RNN), is being used for speech recognition in combination with different deep learning techniques.

2.1.2.2 Mathematical Model

Speech is the expression of ideas and affections by articulated sounds. It is the primary mode of communication between humans and can be safely assumed that people are more comfortable using speech as a communication scheme with machines as opposed to other schemes such as writing.

Automatic speech recognition (ASR) is a software solution that is designed to allow the machine to interpret speech into text, as shown in Figure 2.2. The machine receives the input through the microphone or telephone and converts it into text in the respective language. A typical ASR should be able to perceive the given input, recognize the spoken words and then pass these words as an input to the next module so that action can be performed [23, 24, 25]. Communication between humans is dominated by spoken language, therefore it is natural for people to expect speech interfaces with machines [26].

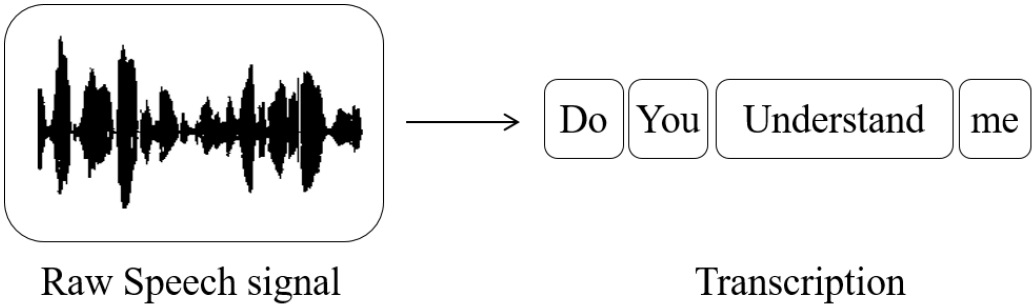

Figure 2.2: Basic Operation of the ASR System

Mathematically, an ASR system is a function $Y = f(X)$ that maps a given input sequence of n audio samples X; $X = \{X_1, X_2, X_3, ..., X_n\}$, from a recorded speech signal, to an output sequence Y; $Y = \{Y_1, Y_2, Y_3, ..., Y_m\}$, of m words. The output sequence Y which consists of words W represents the transcription to the input audio file, and it has the highest posterior probability $P(Y \mid X)$, where $P(Y \mid X)$ is

$$\begin{aligned} W &= argmax\ P(W \mid X) \\ &= argmax\ \frac{P(W)P(X \mid W)}{P(X)} \end{aligned} \tag{2.1}$$

where $P(W)$ is the probability of the occurrence of the word, $P(X)$ is the probability that X is present in the signal, and $P(X \mid W)$ is the probability of the acoustic signal X occurring in correspondence to the word W.

Developing an ASR engine that models this probability is a quite complex task as it should be robust to speaker variations, context, and acoustic environment. For instance, human speech can vary in speed, pronunciation, volume and still result in the same transcription. Mixing such variability with the environmental scenarios such as noise, echo, reverberation, and distance from the

microphone increases the complexity of the task.

The ASR engine typically consists of 4 modules; prepossessing module, feature extraction module, classification model, and language model, as shown in Figure 2.3. The prepossessing model applies noise reduction on the input signal as it is usually noisy. There are a number of filters and methods that can be applied to the input signal to improve its signal-to-noise ratio. Framing, normalization, end-point detection, and pre-emphasis are some of the frequently used methods to reduce noise in a signal [27, 28, 29].

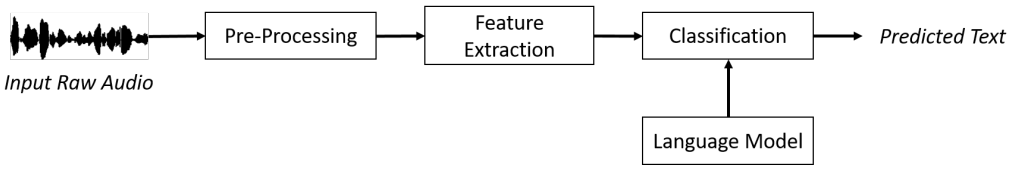

Figure 2.3: Architecture of the ASR Engine

After pre-processing, the clean speech signal is then passed through the feature extraction module. The performance and efficiency of the classification module are highly dependent upon the extracted features [30, 31, 32]. The most commonly used feature extraction methods are Mel frequency cepstral coefficients (MFCCs), linear predictive coding (LPC), and discrete wavelet transform (DWT) [32, 33, 34, 35].

The third module is the classification model which takes the extracted features as an input and predicts the corresponding textual output. There are various approaches to implement such a module. The first is to learn a joint probability distribution from the training set and use that distribution to predict the future output. This approach is called a generative approach; Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) are some common examples. The second approach is to learn model parameters through training on input features and their corresponding output text. Support Vector Machines (SVM) and ANN are the most common examples. Hybrid approaches are also used; HMM-ANN for example [36].

The fourth module is the language model where various rules and semantics of the language are introduced through the incorporation of this model in decoding the output of the ASR. Although recent ASR implementations do not require the use of a language model, they still can enhance the model accuracy significantly.

2.1.2.3 ASR Evaluation Metrics

This section explains in detail the evaluation metrics used to evaluate the performance of an ASR. The performance usually depends on a couple of factors, namely, speed and accuracy.

2.1.2.3.1 Speed

The real-time factor (RTF) is the metric used for calculating the speed of a proposed model. It can be computed using the following formula:

$$RTF = \frac{P}{I} \tag{2.2}$$

where P is the time that the model needs to process the input and I is the duration of the input audio. An RTF of 1 means that the input was processed in "Real-Time". This metric is highly dependent on the hardware infrastructure therefore it has to do with more than just the speed of the ASR model.

2.1.2.3.2 Accuracy

The metric word error rate (WER) is usually used to measure the accuracy of the ASR. It is hard to calculate as the output sequence may not be of the same length as the ground truth sequence. It can be calculated as follows:

$$WER = \frac{S + D + I}{N} \tag{2.3}$$

here S is the number of substitutions in the output as compared to the ground truth. D is the number of deletions, I is the number of insertions and N is the total number of words in the ground truth.

A variation of the WER is the word recognition rate (WRR), which is calculated as:

$$\begin{aligned} WRR &= 1 - WER \\ &= \frac{N - S - D - I}{N} \\ &= \frac{H - I}{N} \end{aligned} \tag{2.4}$$

where $H = N - (S + D)$ represents the total number of correctly guessed words.

2.1.2.4 Benchmark Datasets

This section discusses in detail some of the commonly used datasets for training and benchmarking different ASR designs. It is common in speech-related research communities to evaluate proposed systems based on certain publicly available datasets so that multiple solutions can be fairly compared on the same benchmark. The listed datasets are the formal ones used in the research community and we have used some of them to evaluate our approach.

2.1.2.4.1 LibriSpeech

LibriSpeech corpus [37] is one of the most commonly used corpora for training and benchmarking ASR models. It consists of approximately 1000 hours of 16kHz read English speech, prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from reading audiobooks from the LibriVox project and has been carefully segmented and aligned.

2.1.2.4.2 CSTR VCTK Corpus

The VCTK corpus [38] includes speech data uttered by 109 English speakers with various accents, where each speaker reads about 400 utterances. All recordings were converted into 16 bits, down-sampled to 48 kHz, and manually end-pointed. This dataset contains almost 9h of audio data.

2.1.2.4.3 TIMIT acoustic-phonetic continuous speech Corpus

The TIMIT corpus is an acoustic-phonetic continuous speech corpus [39]. It has recordings of 6300 phonetically rich sentences, read by 630 speakers of eight major dialects of American English. The training set consists of 3.14 h of recording; the rest is divided into the test and development set respectively.

2.1.3 Automatic Speaker Recognition

As shown in Figure 2.4, the term speech signal processing often encompasses a number of functionalities such as recognition, classification, or feature extraction. The main focus of this thesis is recognition.

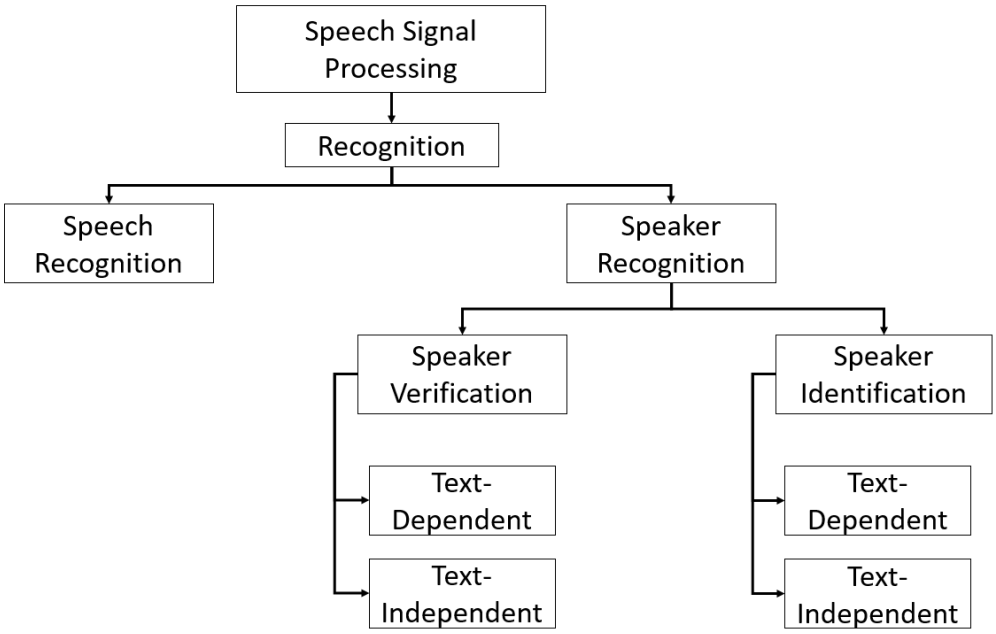

Figure 2.4: Speech Signal Processing Functionalities

Speaker recognition is the other type of recognition performed on speech signals besides speech recognition. This model answers the question of (who is speaking?) as opposed to speech recognition which provides an answer for (what is spoken?). Automatic speaker recognition is the process of recognizing the speaker automatically based on their voice. The recognition task can be used for speaker identification and/or speaker verification.

Speaker verification is where the system matches the claimed identity against a specific speaker's voice model, whereas speaker identification is where the system tries to match an unknown speaker against a group of known voices [40]. Speaker verification/identification is divided into text-dependent and text-independent. Text-dependent systems require providing the same utterance in training and testing (for example, wake-up systems) while text-independent systems don't rely on a specific text.

In the next subsection, we provide a detailed overview of the automatic speaker verification system. We specifically discuss its definition and purpose, introduce the concept of a speaker identity and provide some development history on that, and finally conclude with some evaluation metrics and benchmarking details.

2.1.3.1 Automatic Speaker Verification

Smart assistants provide a hand-free experience for users in which they can control their smart devices and home appliances via speech. As user security is a major concern in such a voice-driven interaction interface, a robust speaker verification module is essentially needed. For example, bank transactions may be verified through a specific sentence uttered by the user to verify his identity. Siri, Google Assistant, and Alexa require a certain keyword uttered by the user's voice. Once verified the devices are configured to respond and access its services.

For the urgent need to develop better speaker verification/identification models, a number of challenges have been held for the speaker recognition task such as the NIST [41] speaker recognition evaluation (SRE), VoxCeleb [42], and the speakers in the wild (SITW) speaker recognition challenge [43].

Automatic speaker verification (ASV) system behavior can be illustrated by the pipeline in Figure 2.5 [2]. Typically, the user is required to record a certain utterance N number of times, "Hey Siri", for example. This is called the enrollment phase where features are extracted from the given data and the user's voice is translated into a speaker model representing the speaker's voice-print and it is saved for reference. At the testing phase, the extracted features will be compared against the enrolled speaker model. The similarity score is calculated between the enrollment speaker and the test speaker. If the score is higher than a predefined threshold then the identity is verified; otherwise, authentication will be rejected. As opposed to that, in speaker identification, the testing speaker is compared with multiple known speaker models to determine the best match.

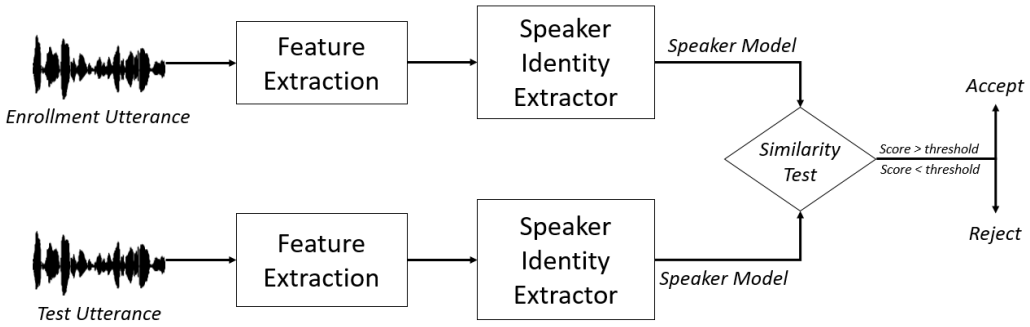

Figure 2.5: Basic Speaker Verification System

2.1.3.2 Speaker Identity

Speaker modeling is essential in the process of speaker verification. In early research, vector quantization (VQ) was used [44] for speaker verification. Then GMMs were proposed for speaker modeling [45]. A GMM is a combination of clusters of probability density functions (PDFs). When a test utterance is given, it can compute the likelihood for each cluster and make a comparison. The cluster with the highest probability is assumed to correspond to the speaker. In GMM, data is modeled as different clusters, each has its own mean vector, weight parameters, and covariance matrix. The likelihood of an utterance is given by:

$$f(x_n \mid \lambda) = \sum_{i=1}^{M} \pi_i \mathcal{N}(x_n \mid \mu_i, \sigma_i) \tag{2.5}$$

where n is the index for a random vector, i is the index for the cluster while M is the number of clusters. μ_i is the mean vector, σ_i) is the covariance matrix, and pi_i is the weight for each component of the GMM.

With the help of GMMs, and to make the speaker verification more general, the universal background model (UBM) was proposed [46], in which the speaker model can be modified based on a background or world model. In GMM-UBM [2], a supervector, which is made by concatenating the parameters of each component, is used as the feature vector for verification. The i-vector approach is just a dimensionality reduction of GMM supervector, which is like principal component analysis (PCA) on GMM supervectors.
With the application of machine learning and deep learning [47] recent research focuses on DNN-based text-dependent verification with "Hello Google". In [48], the concept of x-vector is introduced where the focus is more on text-independent verification. Since then, the x-vector has been used as a baseline in many papers in the ASV field.

The x-vector extractor is a DNN consisting of seven fully connected layers, a stats pooling layer, and a SoftMax output layer, as listed in Table 2.1, where T denotes speech length, t indicates t-th frame, . is a set of frame indices, and N is the number of speakers in training data. This DNN takes 24-dimensional filter banks as input, and the first three layers splice several frames of the previous layer's output as its input. As a result, the third layer can extract one feature vector that covers 15 input frames of the filter bank features. To deal with the varied length of the input speech, a stats pooling layer is used after the fifth layer to calculate the mean and variance of the overall output of the fifth layer. The SoftMax layer predicts the probability that the input speech is from each of the N speakers in the training data. Once the network is trained on a database containing a large number of speakers, the outputs of the higher layers can be used to represent the speaker space, and the trained model can be used to extract the speaker identity for a new input speech signal. Finally,

the utterance-level x-vectors of a speaker are averaged as the speaker-level x-vector.

Table 2.1: Network Architecture of the x-vector System

Layer	**Layer context**	**Context**	**Input X Output**
1	[t - 2; t + 2]	5	120 X 512
2	t - 2; t; t + 2	9	1536 X 512
3	t - 3; t; t + 3	15	1536 X 512
4	t	15	512 X 512
5	t	15	512 X 1500
Stats	[0,T)	T	1500T X 3000
6	0	T	3000 X 512
7	0	T	512 X 512
softmax	0	T	512 X N

2.1.3.3 ASV Evaluation Metrics

The optimal behavior of an ASV system would be to properly identify the enrolled speaker while rejecting all other speakers' attempts. The performance of such a system can be assessed by two metrics. The first is the false acceptance ratio (FAR), which is the percentage of times it falsely accepts an un-enrolled speaker, shown in Equation 2.6. The second metric is the false rejection ratio (FRR), defined in Equation 2.7, which is the percentage of times the system falsely rejects the enrolled speaker. In these equations, FA is the number of false acceptances, FR is the number of false rejections and VA is the total verification attempts.

$$FAR = \frac{FA}{VA} \tag{2.6}$$

$$FRR = \frac{FR}{VA} \tag{2.7}$$

FAR results from accepting the claim of the non-target speaker while FRR results from declining the claim of the target speaker. It is typical that improving the FAR, will typically come at the expense of worsening the FRR and vice versa. Usually, in real applications, ASV systems can be designed by optimizing for the FAR or the FRR in mind based on the system requirements. For example, in applications such as banking verification, designers tend to lower the FAR while increasing the FRR, as the consequences of a false acceptance could be dire. While, in applications such as unlocking the cellphone, designers can lower the FRR while increasing the FAR, as a repeated authentic user rejection will result in a horrible user experience.

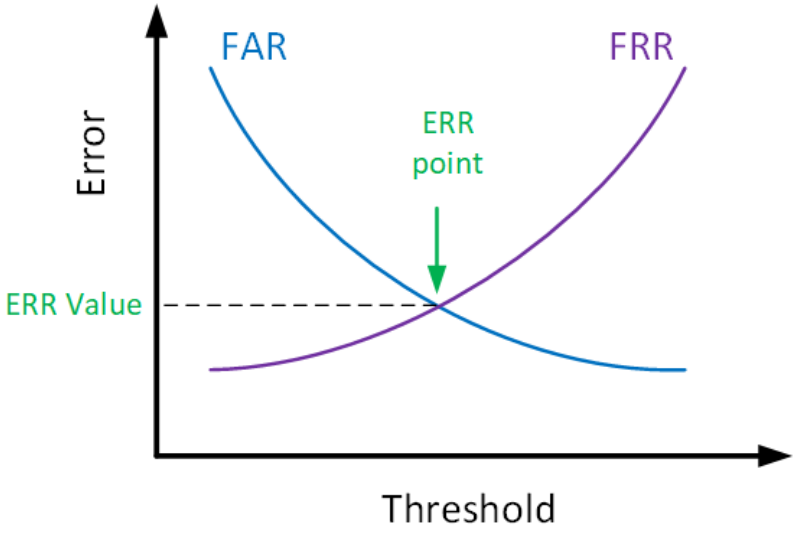

Figure 2.6: Equal Error Rate (EER)

The most common metric used to assess the ASV performance is the equal error rate (EER), also referred to as the crossover error rate [49]. The EER is defined as the error rate at the point when the FAR is equal to the FRR, as shown in Figure 2.6. The use of the EER is very common in biometric security systems.

ASV is one of these biometric systems that aim at verifying the presence of a certain speaker's identity. In these systems false attempts -whether Acceptance or rejection- should be minimal, thus FAR, FRR, and consequently EER should be ideally zero. Low EER systems are systems that can perfectly distinguish the enrolled speaker from all other speakers.

For the case of anonymization, the objective is completely reversed. It is actually desired that the system is able to conceal all of the enrolled speaker identifiable information and replace it with a pseudo-identity, in a manner that this speaker can't be identified anymore using an ASV system enrolled on his voice. Thus, for anonymization systems, the optimal value for EER would be 50%, indicating that the FAR and FRR both equal 50%, which means complete and equal confusion on determining/verifying the speaker identity. Thus, the system is as likely to reject an enrolled user as it is likely to accept an un-enrolled. Any emphasis on FAR or FRR would introduce a bias in the system, which is a design choice.

2.1.3.4 ASV Benchmark Dataset

VoxCeleb [42, 50] is an audio-visual dataset consisting of short clips of human speech, extracted from interview videos uploaded to YouTube. There are 1251 speakers with 153516 utterances in VoxCeleb 1 and 6112 speakers with over one million utterances in VoxCeleb 2.

The main language of VoxCeleb is English, with 37% of speakers from the USA, the others are from the UK, France, India, Germany, etc. In VoxCeleb, 61% of speakers are male speakers while the remaining 39% are female speakers.

2.2 Literature Review on Speech Privacy Preservation

There is limited previous work tackling the problem of speaker anonymization, which motivated the announcement of a challenge in 2020, The VoicePrivacy Challenge [51], to set an initiative spearheading efforts to develop privacy preservation solutions for speech technologies. Challenge organizers developed a baseline system [4] and similar studies emerged as an outcome of the challenge. This section will start by presenting related work in speech privacy preservation. Afterward, an overview of the VoicePrivacy challenge baseline and its outcomes will be discussed.

2.2.1 Speech Privacy Techniques

Generally, the techniques for speech privacy preservation fall into four categories: deletion, encryption, federated learning, and anonymization. Deletion methods [52, 53] are meant for ambient sound analysis where the need is to guarantee a privacy-aware pipeline. acoustic sensors are often deployed in urban environments to passively collect data in public spaces. People in such environments may lack the knowledge of the presence of such sensors, hence their expected privacy might be violated. The deletion method works by separating the linguistic content and personally identifiable voices from the rest of the acoustic scene. They delete or perturb any overlapping speech to the point where no information about it can be recovered.

Encryption methods [1, 54, 55, 56], support computation upon data in the encrypted domain. They convert speech data into an unreadable format that can only be viewed with access to a private key. These approaches trade off efficiency for privacy, resulting in varying overheads of computation, communication, and rounds of interaction. They are also specific to the given application and challenging to integrate with existing systems. As such, cryptographic solutions can be relatively inflexible, unmanageable, and can only be built by specialists.

The idea of federated learning is to collaboratively train a neural network on a server. Each user receives the current weights of the network and in turn, sends parameter updates (gradients) based on local data. This protocol has been designed not only to train neural networks data-efficiently but also to provide privacy benefits for users, as their input data remains on the device and only gradients parameters are shared as in [57]. The derived data used for learning may still leak information about the original data. In [58] they prove that any input to a fully connected layer can be reconstructed analytically independent of the remaining architecture.

Speaker anonymization differs from speech anonymization in that the former suppresses speaker identity while the latter conceals linguistic content.

Speaker anonymization can be done either physically or logically. While physical anonymization aims to perturb speech in physical space by adding an external sound to the original waveform [59], logical anonymization suppresses the speaker identity. Past and recent attempts have focused on noise addition, speech transformation [60], voice conversion [61, 62, 63], speech synthesis [4, 64], or adversarial learning [65].

In [61], a voice transformation (VT) system is presented. It aims to change the speaker identity into another special speaker. Similarly, the approach in [63] utilized a convolutional neural network (CNN) as a VT function and averaged different transformation results to anonymize speech.

In [66] and [62], they improved the convenience of the VT-based method to enable j user to select an approximate transformation from a pool of pretrained VT models for speaker anonymization. Justin et al. [67] performed speaker anonymization by first recognizing the diphones in the input speech using an ASR system and then synthesizing speech from the recognized diphone sequence. The synthesized speech differs from the original one in terms of speaker identity because the synthesizer is speaker-dependent and was trained using the data of a different speaker.

With a goal closely related to that of anonymization, Alegre et al. [68] investigated the so-called speaker evasion and hiding using voice conversion techniques. With the work aiming only to avoid surveillance systems, it evaluated only how the approach could degrade ASV performance. It did not consider degradation to speech quality. In contrast, our proposed system will be evaluated in terms of speaker identity anonymization, speech quality, and linguistic content.

2.2.2 The VoicePrivacy Challenge

The VoicePrivacy challenge, part of Interspeech 2020 special sessions and challenges, is an initiative to spread the efforts towards developing privacy preservation solutions for speech technology. It aims to gather a new community to define the task and metrics and to benchmark initial solutions using the first common datasets and protocols. It takes the form of a competitive challenge. Challenge participants are required to process a dataset of speech signals in order to anonymize them while protecting the linguistic content and speech naturalness. The challenge ran from early 2020 and concluded with a special session held in conjunction with Interspeech 2020.

The main objective of the VoicePrivacy challenge is to encourage progress in the development of anonymization techniques for speech data. The specific technical goals as shared in the challenge evaluation plan [5], are summarized as follows:

- Develop novel techniques to suppress speaker-discriminative information within speech signals.
- Promotes effective anonymization techniques while protecting intelligibility and naturalness;
- Provide benchmarking techniques to facilitate the comparison of different anonymization solutions using a common dataset and protocol;
- Investigate metrics for the evaluation and meaningful comparison of different anonymization solutions.

The anonymization method in [4], shown in Figure 2.7, comprises of three steps: (1) extraction of x-vector [48], pitch (F0) and bottleneck (BN) features; (2) x-vector anonymization; (3) speech synthesis (SS) from the anonymized x-vector and the original F0 and BN features. The baseline is designed under the assumption that the information in a speech waveform can be classified into two groups: one group mainly encodes the speech content such as the sequence of phonemes, while the other group captures the acoustic features invariant to the speech content, i.e., the speaker identity. Therefore, a speech waveform can be anonymized by altering the features that encode the speaker identity only.

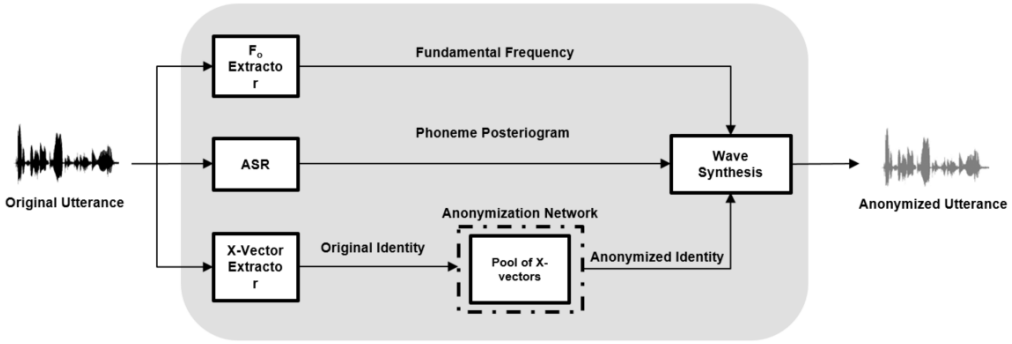

Figure 2.7: Speaker Anonymization Pipeline [4]

The system first extracts an x-vector, a phoneme posteriogram (PPG), and fundamental frequency (F0) from the input waveform. It then anonymizes the x-vector on the basis of a pool of x-vectors of external speakers. An x-vector is composed based on a similarity score s. The distance measure can be either PLDA distance or cosine distance. The final x-vector can be calculated by averaging a set of candidate x-vectors for which the similarity to the original speaker is in the range $[s - \epsilon\ ,\ s + \epsilon]$, where $\epsilon > 0$ is a hyper-parameter used to control the width of the range (denoted as "nearest" or "farthest"). Finally, it uses an acoustic model and a neural waveform model to synthesize the speech waveform using the anonymized x-vector and the original PPG and F0.

The system in [4] suffers from inaccurate phoneme posteriogram representations which provide poor linguistic depictions as well as high WERs for largely distant anonymized x-vectors. This indicates that when x-vectors are averaged over very different unseen speakers, the system is unable to correctly recover the original linguistic contents of the input speech.

Two relevant studies emerged as an outcome of the VoicePrivacy Challenge in [69] and [70]. Both of them are based on the baseline system presented in 2.2.1. Turner et.al [69] proposed a distribution-preserving voice anonymization technique. Starting from an observation on the baseline in which the anonymized x-vectors are similar to each other due to averaging, hence losing much information, the proposed a new method to generate fake x-vectors which tackle these issues by approximating the original distribution of x-vectors and their intra-similarities. Using their generative model, they got rid of the pool and use the rest of the baseline components to produce anonymized speech.

Their method can be summarized as follows: they applied PCA on x-vectors obtained from a

large dataset, thus reduced the space of the vectors in order to sample from it, then fitted a generative model, i.e. GMM, to generate reduced-dimensional x-vectors that could be brought back to the 512-dimensional space by applying inverse PCA transform.

When an utterance needs to be anonymized, the GMM is randomly sampled to produce a fake x-vector. To mitigate the risk of the fake speaker being close to the original one, an optional forced dissimilarity measure is used to ensure the distance between the two x-vectors is above a certain threshold.

Reported results on EER show heavy alternation in values, being sometimes better than the baseline and other times worst. It is fair to note that all EER values fall within an acceptable range indicating the difference in identities before and after anonymization. However, WER results on all test sets are higher than the baseline measures, indicating further degradation in the preservation of linguistic content.

Champion et. al [70] modifies the baseline by including f_o in the anonymization process. They analyzed the impact of this modification across gender and found that it can always improve the anonymization process.

They modified f_o using the following linear transformation:

$$F = \mu_y + \frac{\sigma_y}{\sigma_x}(x_t - \mu_x) \tag{2.8}$$

where F represents the log-scaled f_o of the source speaker at the frame t, μ_x and σ_x represent the mean and standard deviation for the source speaker. μ_y and $\sigma_y$represents the mean and standard deviation of the log-scaled f_o for the pseudo-speaker.

Chapter 3

The Proposed Speaker Anonymization Model

A typical speaker anonymization system outputs a speech waveform while hiding all personally identifiable information and maintaining naturalness and linguistic content. The baseline system relied on a pool of x-vectors of external speakers to sample from and create a new x-vector. In this thesis, an alternative anonymization approach is proposed through the development of a generative adversarial network (GAN) that is able to generate plausible x-vectors. In this chapter, a comprehensive discussion on GANs is provided, then the design of the proposed system which uses different GAN alternatives in the generation of x-vectors is presented.

3.1 Generative Adversarial Networks

Discriminative models usually map high-dimensional rich sensory input to a class label [71, 72]. Such models are primarily based on the backpropagation and dropout algorithms, using piece-wise linear units [73, 74, 75] which have a particularly well-behaved gradient. Whereas deep generative models try to model the distribution of the training data.

A GAN [76] is a class of machine learning frameworks where two neural networks; the generative network and the discriminative network, compete against each other in a game. Figure 3.1 shows the general structures of GANs. The generative network generates candidates while the discriminative network evaluates them. The contest operates in terms of data distributions. Typically, the generative network learns to map from a latent space to a data distribution of interest, while the discriminative network distinguishes candidates produced by the generator from the true data

distribution. The objective of the training in generative networks to increase the error rate of the discriminative network i.e., "fool" the discriminator network by producing fake candidates which look similar to the real samples.

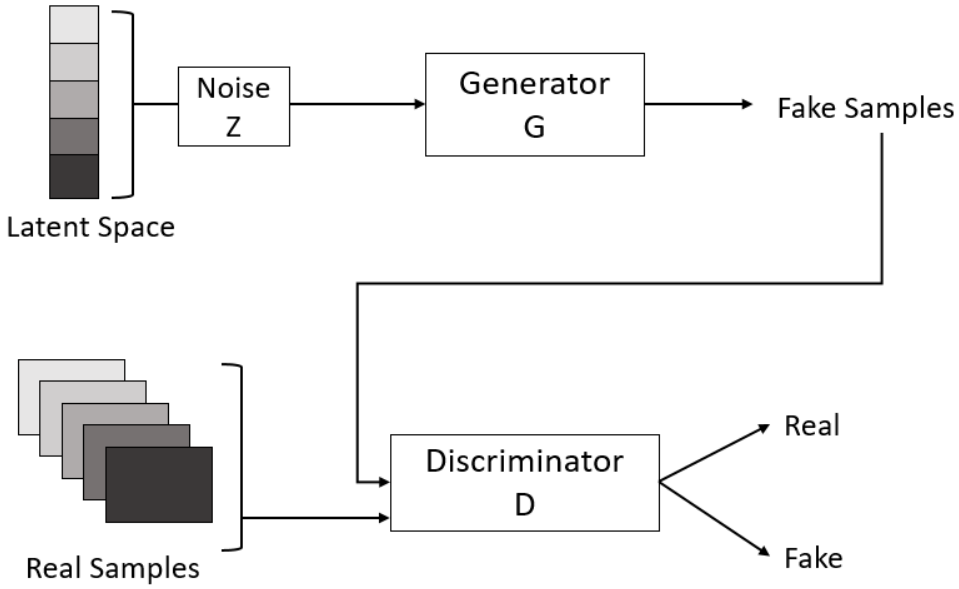

Figure 3.1: Generative Adversarial Network Architecture

A known dataset serves as the initial training data for the discriminator. Training the discriminator involves presenting it with samples from the training dataset until it achieves acceptable accuracy. The generator is trained to fool the discriminator. Typically, the generator is seeded with randomized input that is sampled from a predefined latent space (e.g., a multivariate normal distribution). Thereafter, candidates synthesized by the generator are evaluated by the discriminator. Backpropagation is applied in both networks so that the generator produces better samples, while the discriminator becomes more skilled at flagging synthetic samples.

Many GAN designs have been proposed in the literature. Each design is tuned to meet certain application requirements. As compared to the rich GAN literature in generating images (human faces and animals), this thesis is concerned with the investigation of generating audible human voices.

The idea of a generative adversarial network was first introduced in 2014, where the authors of [76] proposed a new framework for estimating generative models via an adversarial process. In this

framework, they train two models simultaneously; a generator model G that captures the distribution of the data and a discriminator model D that estimates the probability that a sample belongs to the training data rather than the generated data.

The purpose of the training scheme for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. A minimax game is a decision rule used for minimizing the possible loss for a worst-case (maximum loss) scenario or maximizing the minimum gain. The authors defined G and D by multi-layer perceptrons where the entire system can be trained with backpropagation.

As highlighted in Figure 3.1, the generative model competes against the discriminative model. The former tries to mimic the training data by learning its distribution while the latter learns to determine whether a sample is from the model distribution or the data distribution. Competition in this game drives both models to improve their methods until the generated fake samples are indistinguishable from the real samples.

To learn the generator's distribution p_g over data x, a prior on input noise variables p_z(z) is defined, then a mapping is represented to data space as G(z; θ_g), where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. A second multilayer perceptron D(x; θ_d) that outputs a single scalar was defined. D(x) represents the probability that x came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize log(1 D(G(z))). Typically, D and G will compete against each other following the value function $V(G, D)$:

$$\min_G \max_D V(D,G) = E_{x\sim p_data(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z)))] \tag{3.1}$$

The training scheme for a GAN takes place in an alternating manner; where we alternate between k steps of optimizing D and one step of optimizing G.

Goodfellow et al. [76] showed that, for a fixed generator, there is a unique optimal discriminator:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{3.2}$$

where p_{data} is the PDF of the training data and $p_g(x)$ is the PDF of the generated data.

They also showed that the generator, G, is optimal when p_g (x) =p_{data} (x), which is equivalent to the optimal discriminator predicting 0.5 for all samples drawn from x. In other words, the generator is optimal when the discriminator cannot distinguish real samples from fake ones.

Despite the theoretical existence of unique solutions, GAN training is challenging and often unstable for several reasons [77, 78, 79]. These challenges include:

- Convergence difficulties, where the point of equilibrium between discriminator and generator is not found. A straightforward indication of this is the discriminator loss approaching zero. This mode is usually caused by the generator generating corrupted data that can be easily classified into fake by the discriminator [77].
- Mode collapse, which is the case when the generator gets stuck in producing similar samples for different inputs [78].
- Vanishing gradients in the discriminator network, providing no reliable path for gradient updates to the generator [79].

This GAN design, also known as vanilla GAN, was benchmarked on relatively simple image data sets: MNIST (handwritten digits), CIFAR-10 (natural images), and the Toronto Face Data Set (TFD).

Followed that, a work introduced in [80] called conditional GAN (CGAN) where conditioning ability is added to the GAN. Typically, the conditioning happens in the form of a label fed to both generator and discriminator. As compared to equation (3.1), the objective function in a CGAN is:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x \mid \mathbf{y})] + E_{z\sim p_z(z)}[log(1 - D(G(z \mid \mathbf{y})))] \tag{3.3}$$

The Wasserstein Generative Adversarial Network (WGAN) [81], is an extension to the generative adversarial network, introduced in 2017. This GAN type improves the training stability and provides a loss function that correlates with the quality of generated images. It is an important extension to the GAN model and requires a conceptual shift away from a discriminator that predicts the probability of a generated image being "real" and toward the idea of a critic model that scores the "realness" of a given image. This conceptual shift is motivated mathematically using the earth mover distance, or Wasserstein distance, to train the GAN that measures the distance between the data distribution observed in the training dataset and the distribution observed in the generated examples.

In 2019, a study on modeling the PDF of rows in a tabular took place in [82], named conditional tabular GAN (CTGAN). The study discussed the difficulties in modeling tabular data, such as the mixture of discrete and continuous columns, and continuous columns may have multiple modes whereas discrete columns are sometimes imbalanced making the modeling difficult.

The authors introduced several new techniques to mitigate those difficulties. Techniques included: augmenting the training procedure with mode-specific normalization, architectural changes, and addressing data imbalance by employing a conditional generator and training-by-sampling.

Several unique properties of tabular data are handled in the CTGAN design. Such properties include the presence of mixed data types, non-gaussian distributions, and multimodal distributions. Authors of [82] used mode-specific normalization to overcome the non-Gaussian and multimodal distribution and designed a conditional generator and training-by-sampling to deal with the imbalanced discrete columns.

In this work, while the x-vectors are considered to be continuous numerical vectors that don't contain mixed data types nor multimodal distribution, it is a non-gaussian distribution, which means that its values are not bounded by [-1,1] and we can't use tanh function in the last layer of the network because that will lead to a vanishing gradient problem.

3.2 Anonymization Component Design

In this section, a comprehensive overview of the proposed anonymization module is presented. First, the motivation behind using a generative model is explained. Next, the process of selecting a suitable generative model design is discussed.

3.2.1 System Architecture

In the baseline discussed in Section 2.2.2, the system relied on a pool of original speaker identities, i.e. x-vectors, and applied some distancing and averaging techniques to create a pseudo-identity to anonymize the input utterance. Such a technique poses a serious risk of using an already existing human identity in the anonymization process or an identity close to it. This may protect the privacy of one user at the expense of violating the privacy of another user as the pool is comprised of real human identities.

This motivated the addition of another level of anonymity by using a GAN for generating x-vectors for the pool as shown in Figure 3.2. A GAN that is able to infer pseudo identities that are natural yet unique and irreversible as the network will be optimized for minimizing the difference between the real and the fake x-vectors.

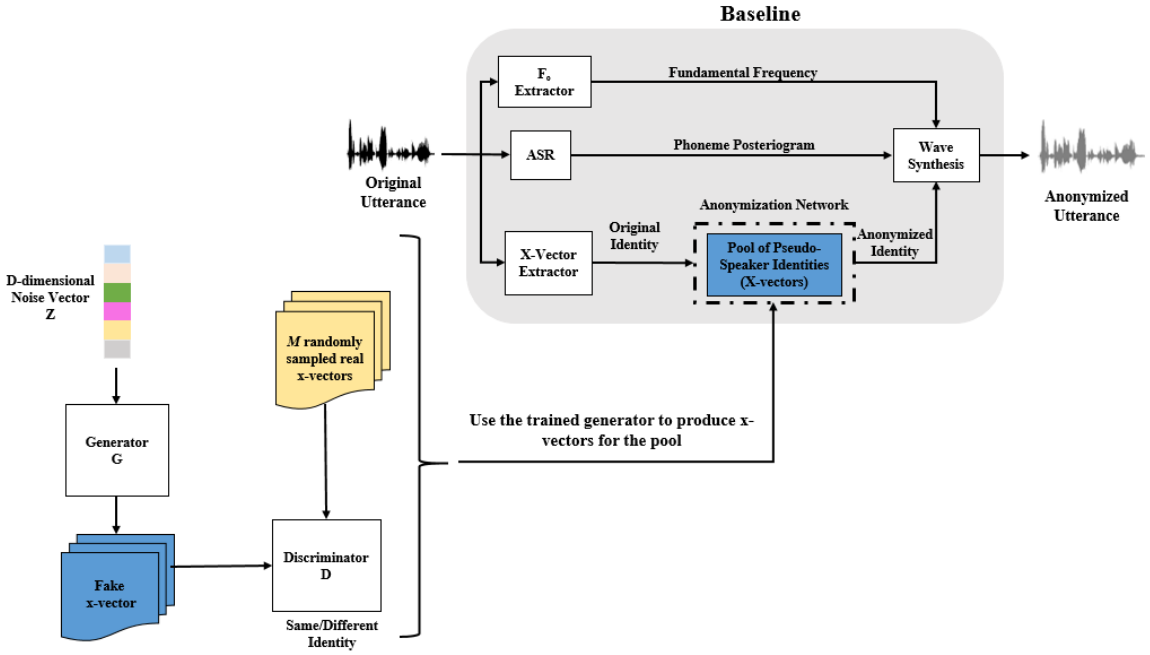

Figure 3.2: Proposed Anonymization System Design

The system in Figure 3.2 performs anonymization in three main steps:

1. **Feature Extraction:** extract the speaker x-vector, the fundamental frequency (F0) and bottleneck (BN) features from the original audio waveform.
2. **Speaker Anonymization:** anonymize the x-vector of the source speaker using an external pool of speakers. In this work, this pool is created using a generative model, as opposed to the baseline in [4] which uses libriTTS corpus to construct the pool.
3. **Speech Synthesis:** synthesize the speech waveform from the anonymized x-vector.

The proposed generative model is trained to generate x-vectors. As discussed in section 2.1.3.2, an x-vector is a 512-dimensional vector that represents the speaker's identity in speech. A classification model is trained on a dataset of different speakers with the final goal being to predict the speaker identity. Raw input speech is processed to obtain 30-dimensional MFCCSs with a frame length of 25ms, mean-normalized over a sliding window of up to 3 seconds. A speech activity detection model SAD is used to filter out nonspeech frames. A time-delay neural network is trained to predict the speaker identity given an input speech.

The network is trained to classify the N speakers in the training data. A training example consists of a chunk of speech features (about 3 seconds average), and the corresponding speaker label. After training, embeddings are extracted from layer segment6. Excluding the SoftMax output layer and segment7 (because they are not needed after training) there is a total of 4.2 million parameters. In this work, a pre-trained model that is part of *Kaldi* speech recognition toolkit [83] was used for x-vector extraction.

3.2.2 GAN Selection

Four GAN types are considered in this work; vanilla GAN, CGAN, WGAN, and CTGAN. The first three are the most popular GAN designs in the image domain and the latter is a choice suitable for tabular data; x-vector in our case. Table 3.1 presents the architecture and hyperparameters used in implementing each of the four GAN designs.

The methodology adopted in designing the architecture for GAN, CGAN, and WGAN is to reflect the network behavior from the image domain to deal with a 512-dimensional vector, doing all necessary changes in network size, activation functions, and layers type. For instance, a convolutional network was an overkill for the task and caused an unstable GAN training, causing the discriminator to dominate the generator quickly. The choice of a fully connected dense layer came as a simplification attempt to the network, as it is supposed to model the distribution of vectorized data that is generated by another neural network, i.e. x-vectors, thus are considered structured data. Unlike images, x-vectors don't have strong spatial dependencies, and unlike time-series waveforms, x-vectors are not sequential data.

Each of these networks is to be trained to generate a pool of anonymous x-vectors that will replace the real pool of x-vectors in the baseline. Grid search was performed and the best parameters were chosen and presented in 3.1. The network with the best performance in terms of generalization ability is to be adopted. Quality assessment through a number of metrics and experimental results are provided in Chapter 4.

Table 3.1: The Four GAN Designs' Architectures and Hyperparameters

Model	**Generator**	**Discriminator**	**Batch**	**Epoch**
GAN	Dense(128,activation='relu') BatchNormalization Dense(512,activation='relu') BatchNormalization Dense(512,activation='relu') BatchNormalization	Dense(1,activation='relu') BatchNormalization Dense(1,activation='sigmoid')	128	30000
WGAN	Dense(512,activation='relu') Dense(256,activation='relu') Dense(512,activation='relu') BatchNormalization	Dense(512,activation='relu') Dense(128,activation='relu') Dense(1)	128	30000
CGAN	Dense(256,) LeakyReLU(alpha=0.2) BatchNormalization() Dense(512) LeakyReLU(alpha=0.2) BatchNormalization	Dense(512,) LeakyReLU(alpha=0.2) Dense(1)	500	20000
CTGAN	Dense(256,) LeakyReLU(alpha=0.2) BatchNormalization Dense(256,) LeakyReLU(alpha=0.2) BatchNormalization Dense(512,activation='tanh') Dense(512,activation='gumble') Dense(512,activation='gumble')	Dense(256,) LeakyReLU(alpha=0.2) Dropout Dense(256,) LeakyReLU(alpha=0.2) Dropout Dense(1)	100	1000

Chapter 4

Results and Discussion

In this chapter, different variations of generative adversarial networks are investigated. Evaluation is based on four metrics, each of which assesses one of the desired design outcomes. The environment setup that is used to execute all experiments is first presented. Then, the experimental results for evaluating both the generative models and the anonymization quality are presented and discussed.

4.1 Experimental Setup

Experiments were performed on a Lenovo IdeaPad L340 Gaming 9Gen Intel Core i7 4.5GHz 12M Cash 6-Cores, 8GB RAM, 256GB SSD +1 TB HDD, Nvidia GTX 1650 4GB, and Ubuntu 20.04 operating system.

All experiments were based on the challenge publicly available baseline [4]. Software packages involved: Python 3.6 for training the GAN models, Kaldi toolkit [83] for the pre-trained models. The GAN, CGAN, and WGAN were implemented using Keras python framework, whereas the CT-GAN was implemented using PyTorch framework. The training, development and evaluation sets are those discussed in Section 4.3. The pool of external speakers on which x-vectors are computed to train the GAN model is LibriTTS train-other-500 and VoxCeleb1,2. Additional information on the number of speakers and the gender distributions can be found in Table 4.2.

4.2 Evaluation Methods

The performance of the proposed anonymization system is evaluated using four different measures; cross-cosine similarity distribution, Kolmogorov-Smirnov (KS) score, EER, and WER. The first two

metrics are used to evaluate the quality of x-vectors generated by the GAN while the latter two are for evaluating the speaker anonymization quality. In the following two subsections, we will first discuss cross-cosine similarity distribution as a metric and KS scores for different GAN designs, then we will discuss EER and WER measures after anonymization took place.

4.2.1 Generative Model Evaluation

As discussed in Chapter 3, the literature on GAN design is very rich, containing tens of GAN types each tailored to a specific application. Very few studies tackled the use of GANs in speech as opposed to the enormous number of research that took place in images. This fact set an uncharted territory for us to explore in many aspects. First, designing a GAN that converges for the task of generating speaker identities i.e. x-vectors which are 512-dimensional vectors, and second, evaluating the generation quality of such raw embeddings.

GAN evaluation is often challenging. This challenge comes from the fact that, unlike other deep neural networks (DNN) models that are trained with a loss function until convergence, a GAN generator is trained through the feedback given from another model, the discriminator, that is trained to classify fake from real data samples. The generator and discriminator models are trained together in an alternating and dependent manner. As a result of this, there is no formal objective loss function used to train the generator and therefore, no objective way to monitor the progress of the training and the quality of the generated data.

In the case of image-based GANs, the most straightforward and revealing inspection method would be to directly evaluate the resulting images, while in our case, that's quite not possible. The nature of x-vectors is a DNN-based embedding vector that has certain numerical bounds, nevertheless can't be judged unless incorporated with some phonetic content and synthesized back to an audible waveform to be heard and judged. That's obviously a very complex approach and time-consuming for evaluating a generative model and an approach that involves a number of cascaded models, which means the possibility of introducing a cascaded error that will compromise the whole evaluation process.

On top of that, one of the most common problems in training a GAN is mode collapse, where the generator gets stuck in producing one or a few distinct samples rather than capturing the whole data distribution of the training data and producing diverse samples within that distribution. This usually happens due to the discriminator failing in giving proper feedback on the spread of the generated images, as a result of the basic loss function used in the original GAN [76] that was enhanced in later GAN designs.

Based on the nature of the x-vectors and the desired objective from the GAN generator, we evaluated each GAN design based on two metrics to assess the generation quality. Cross-Cosine Similarity and KS statistic score.

The cross cosine similarity, defined as :

$$\cos(\theta) = \frac{\sum(a_i b_i)}{\sqrt{\sum a_i^2}\sqrt{\sum b_i^2}} \tag{4.1}$$

where a and b are two vectors, was used to assess the presence of mode collapse and to examine the nature of the generated samples' distribution as opposed to the real samples where the cross-cosine similarity measure was computed among different x-vector samples. Generated/real data is shuffled, and cross cosine distance is calculated between pairs of samples to formulate a distribution of similarity among the data. A distribution centered above 0.9 (number derived from experimenting on LibriSpeech random subset, evaluating the cosine distance for the same speaker x-vector in different acoustic condition, similarity above 90% indicated the same speaker identity) would indicate that most samples are similar to each other i.e. represent quite the same speaker, which would be a result of the GAN entering a mode collapse. We used this metric to rule out GAN designs that suffer from mode collapse. For anonymization purposes, this would mean producing few distinct speaker identities (x-vectors).

The choice of using the Cosine distance instead of the Euclidean distance is influenced by the high dimensionality of the data; each x-vectors is compromised of 512 values. The cosine distance measures the orientation between two vectors, while the Euclidean distance measures the direct distance between the end of the two vectors. For high dimensional data, cosine similarity reveals better similarity insights. This is in addition to being a bounded value, which facilitates the analysis.

On the other hand, the KS statistic score is used to determine how close two distributions are to each other, in this work, the fake distribution is to the real one. This metric is computed for each GAN design as well. GANs with greater KS score are the ones which didn't converge to the objective of modeling the distribution of real data i.e. their synthesized identities doesn't follow the distribution of the real identities, thus are ruled out from the possible candidates to use in anonymization.

4.2.2 Speaker Anonymization Evaluation

To assess the quality of anonymization, two systems are used; ASV and ASR. These systems are used for evaluation purposes in this thesis, hence the referred to as ASV_{eval} and ASR_{eval}, respectively).

As discussed in 2.1.2 and 2.1.3, ASV is the process of authenticating individuals through speech and ASR is the process of transcribing time-based speech waveforms into textual format. These are considered objective *privacy* metrics to assess speaker re-identification and linkability in addition to preserving the linguistic content.

Suppose a speaker has been enrolled in the ASV_{eval} system i.e. the system recognizes him through identifying his unique voice footprint. Ideally, the anonymized speech of the speaker will be 'falsely' rejected by the ASV_{eval} system. By calculating how the rejection rates **rise** over multiple speakers, we evaluated how well the proposed system anonymized the speech waveforms. We used the ASR_{eval} system to recognize the word sequences from the original and anonymized speech. It is assumed that the **smaller** the difference between the word sequences, the better the preservation of the linguistic contents. We used the EER to measure anonymization performance and the WER to investigate how well the content was preserved in the anonymized speech [5].

An alternative performance measure to EER is the log-likelihood-ratio cost function (C_{llr}), proposed in [12] as an application-independent evaluation objective. C_{llr} can be decomposed into a discrimination loss (${C_{\text{llr}}}^{min}$) (how good are two classes separated for any threshold) and a calibration loss ($\text{C}_{\text{llr}} - {C_{\text{llr}}}^{min}$) which represents LLRs in the case of C_{llr} being less than 1.
The speaker verifiability metrics, based on the ASV_{eval} which produces log-likelihood ratio (LLR) scores, are applied as follows:

1. Compute PLDA (LLR) scores for (a) clean enrollment data and (b) anonymized trial data;
2. Compute PLDA (LLR) scores for (a) anonymized enrollment data and (b) anonymized trial data;
3. For steps 1 and 2, calculate equal error rate (EER) and log-likelihood-ratio cost function (C_{llr}) [84].

As stated by the challenge in their baseline [4] work, the x-vector-PLDA-based ASV_{eval} [85], as implemented in Kaldi speech recognition toolkit [83], was trained on the VoxCeleb dataset and then adapted to the VCTK domain using 2580 utterances from 20 unused speakers in the VCTK corpus. Finally, the ASR_{eval} module was DeepSpeech [86] pre-trained on external clean data.

Experiments were conducted on the GANs that showed plausible results in the first two metrics. Despite the fact that the GAN can produce an infinite amount of data, We randomly sampled (1k, 3k, and 5k) samples from each GAN to form the pool in an attempt to study the effect of the pool size on the anonymization quality.

The method of identity selection is the same as in [4], where an x-vector is composed based on a similarity score. The distance measure can be either PLDA distance or cosine distance. An experiment was conducted on changing the distance choice to study its effect on the anonymization process.

4.3 Data Preprocessing

A number of publicly available datasets are used for training, development, and evaluation of speaker anonymization systems. They are comprised of subsets from the following corpora:

- **LibriSpeech** [37] is an English speech corpus derived from audiobooks and designed for ASR research. It consists of approximately 1,000 hours of speech sampled at 16 kHz.
- **VCTK** [38] is an English speech corpus of 109 native speakers with various accents. It contains about 44 hours of speech sampled at 48 kHz.
- **LibriTTS** [87] is an English speech corpus derived from the original LibriSpeech corpus and designed for TTS. It contains approximately 585 hours sampled at 24 kHz.
- **VoxCeleb-1,2** [42, 50] is an audiovisual corpus extracted from videos uploaded to YouTube and designed for speaker verification research. It contains about 2,770 hours of speech (16 kHz) from about 7,360 speakers, covering a wide range of accents and languages.

A detailed description of the data used for training, development, and evaluation is given in the following subsections.

4.3.1 Training Data

This anonymization system was built using five different models, where we only trained the GAN model and used the rest of the models as pre-trained models. Details for training these components are presented in Table 4.1.

Table 4.2: Statistics of the Training Datasets of the x-vector Anonymization Model.

Subset	Size (hr)	Number of Speakers			Number of Utterances
		Female	Male	All	
VoxCeleb-1,2	2,794	2,912	4,451	7,363	1,281,762
LibriTTS train-other-500	310	560	600	1,160	205,044

Table 4.1: Anonymization System: Model Description and Training Corpora.

#	Model	Architecture	Input	Output	Training Data
1	ASR AM	TDNN-F 7 TDNN-F hidden layers softmax: 6032 triphone ids LF-MMI and CE criteria	$MFCC^{40}$ + $i-vectors^{100}$	BN^{256} features	Librispeech: train-clean-100 train-other-500
2	X-vector Extractor	TDNN 7 hidden layers + 1 stats pooling layer 7232 speaker ids CE criterion	$MFCC^{30}$	Speaker $x-vectors^{512}$	VoxCeleb: 1, 2
3	Speech Synthesis AM	Autoregressive (AR) network FF * 2 + BLSTM + AR + LSTM * 2 + highway-postnet MSE criterion	$F0^{1}$ + $BN^{256} + x-vectors^{512}$	$Mel-filterbanks^{80}$	LibriTTS: train-clean-100
4	NSF model	sinc1-h-NSF in [88] STFT criterion	$F0^{1} + Mel-fbanks^{80}$ + $x-vectors^{512}$	speech waveform	LibriTTS: train-clean-100
5	X-vector GAN	Conditional Tabular Generative Adversarial Network CTGAN	$Noise-Vector^{100}$	Pool of speaker x-vectors	LibriTTS-train-other-500 VoxCeleb1,2

Vox-Celeb1,2 & LibriTTS train-other-100 corpora were used to train our anonymization model, i.e. GAN model. A more detailed description of the data is provided in Table 4.2.

4.3.2 Development Data

Development data also called the holdout set or validation set, is usually used to evaluate different model designs and tune the hyperparameters for the best values before the final and formal evaluation on the testing set.

As stated by the challenge organizers in their evaluation plan [5], anonymized utterances are referred to as trial utterances, while enrollment utterances are several utterances for each speaker, which may or may not have been anonymized. The attacker is assumed to have access to various amounts of data; one or more trial utterances and possibly, several enrollment utterances for each speaker. The attacker has no access to the anonymization system though.

The level of protection of personally identifiable information is assessed through a range of *privacy* metrics that include objective speaker verifiability metrics that assume the following attack scenario: the attacker has access to a single anonymized trial utterance and several enrollment utterances. Two sets of metrics will be computed, corresponding to the two situations when the enrollment utterances are cleanly anonymized. In the latter case, it is assumed that utterances have been anonymized in the same way as the trial data using the same anonymization system, i.e., all enrollment utterances from a given speaker are converted into the same pseudo-speaker, and enrollment utterances from different speakers are converted into different pseudo-speakers. The validation set is split into a trial subset and an enrollment subset.

Table 4.3 highlights some details about the validation datasets. For the LibriSpeech-dev-clean dataset, the speakers in the enrollment set are a subset of those in the trial set. For the VCTK-dev dataset, two subsets were created of trial utterances, denoted as common part and different part. Both include trials from the same set of speakers but from disjoint subsets of utterances. The common part of the trials is composed of utterances $\# 1 - 24$ in the VCTK corpus, which is identical for all speakers: the elicitation paragraph6 (utterances $\# 1 - 5$) and rainbow passage7 (utterances $\# 6 - 24$). The enrollment subset and the different parts of the trials are composed of distinct utterances for all speakers (utterances with indexes ≥ 25).

Table 4.3: Statistics of the Validation Datasets [5].

Dataset	Subset	Female	Male	Total
Librispeech: dev-clean	Speakers in enrollment	15	14	29
	Speakers in trials	20	20	40
	Enrollment utterances	167	176	343
	Trial utterances	1,018	960	1,978
VCTK-dev	Speakers (same in enrollment and trials)	15	15	30
	Enrollment utterances	300	300	600
	Trial utterances (common part)	344	351	695
	Trial utterances (different part)	5,422	5,255	10,677

4.3.3 Evaluation Data

Similar to the development data, the test subsets from two different corpora (LibriSpeech and VCTK) are used for evaluation. Those datasets are split into enrollment and trial subsets as summarized in Table 4.4.

Table 4.4: Statistics of the Test Datasets [5].

Dataset	Subset	Female	Male	Total
Librispeech: test-clean	Speakers in enrollment	16	13	29
	Speakers in trials	20	20	40
	Enrollment utterances	254	184	438
	Trial utterances	734	762	1496
VCTK-test	Speakers (same in enrollment and trials)	15	15	30
	Enrollment utterances	300	300	600
	Trial utterances (common part)	346	354	700
	Trial utterances (different part)	5,328	5,420	10,748

4.4 Experimental Results

This section outlines evaluation results and discussion for the evaluation methods discussed in 4.2 on the datasets specified in 4.3. Exact results are available in Appendix A.

4.4.1 Generative Model Evaluation Results

Figure 4.1 shows the KS results for the different types of GANs and genders. The KS metric measures the approximate distance between the fake data Probability Distribution Function (PDF) and the real data PDF. The results rule out the CGAN as there is high similarity between the fake and original distributions. However, the KS values are are inconclusive for the GAN, WGAN, and CTGAN.

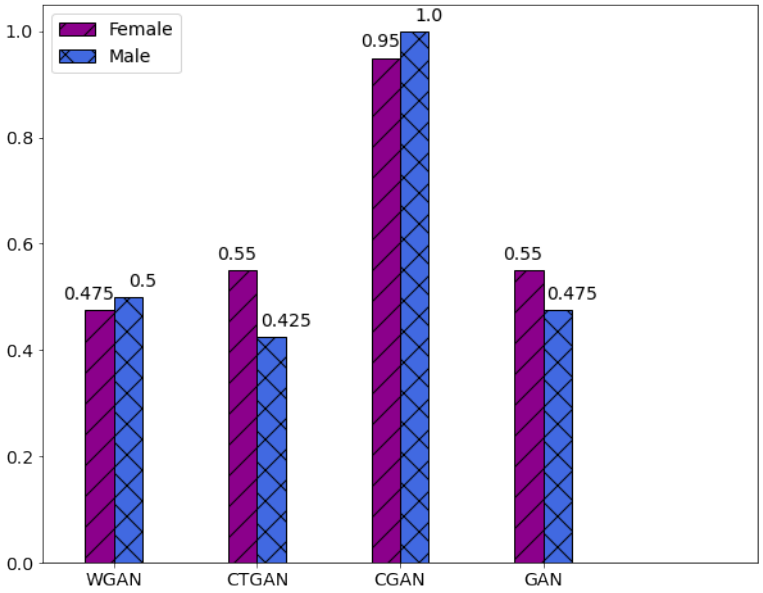

Figure 4.1: KS Test for Each GAN Design Generated Data Against Real Data for Each Gender.

Accordingly, the cross-cosine similarity measure was used to further assess the diversity of the generated x-vectors by computing the cosine similarity between x-vector pairs. Figure 4.2 compares the cross-similarity distribution for the four different GAN designs. When compared to the distribution of the original data for both males and females, shown in Figure 4.2, the x-vectors generated by the vanilla GAN appears to have a slightly similar distribution but with some undesired density above 0.8. On the other hand, the CTGAN's data distribution is good enough to indicate diverse identities. While in the case of a WGAN and CGAN they show a dominating state of mode collapse.

Hence, this leaves the GAN and CTGAN as potential candidates that learned the PDF of real x-vectors. In the next section, the anonymization performance of these two models is performed.

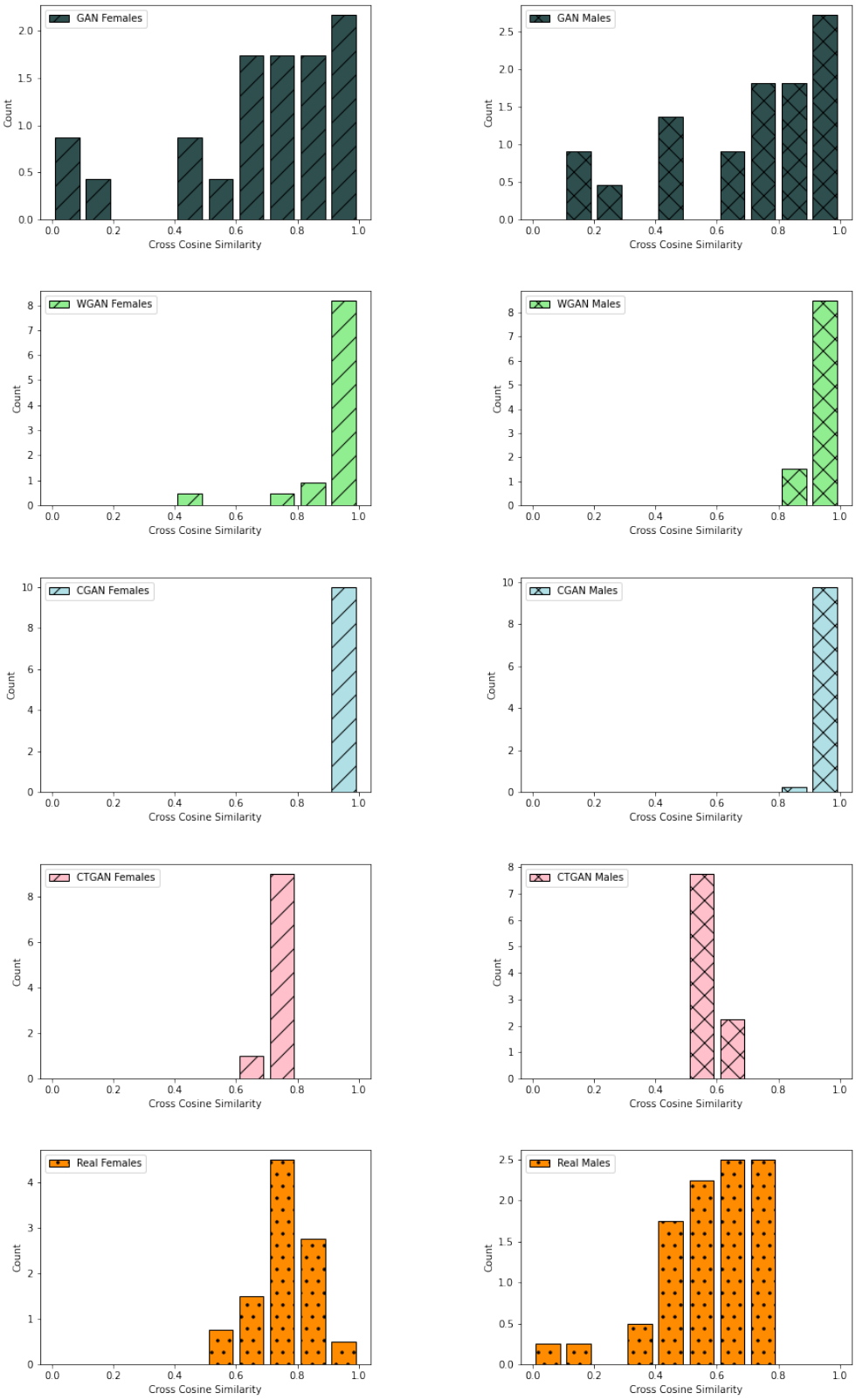

Figure 4.2: Cross-Cosine Similarity Distribution for Four GAN Designs (CTGAN, CGAN, GAN, WGAN) Against the Original Data.

4.4.2 Speaker Anonymization Evaluation Results

In this section, the results of several experiments are presented. These include the anonymization performance of the GAN and CTGAN when varying the pool size and the distance measure. In addition, the proposed anonymization system is compared against two other anonymization systems.

4.4.2.1 GAN Pool Evaluation

Figure 4.3 and Table 4.5 show the anonymization evaluation results for the GAN-generated pool. The Pool size here is 1k. While ASV evaluation results are generally comparable to the baseline and CTGAN and worse in some test sets, the WER results are extremely high. Effectively, more than 90% of the transcribed anonymization utterance is wrong compared to the same non-anonymized utterance. This indicates the poor quality of the anonymized waveform and the failure to maintain the linguistic content.

Note that a language model LM is a probability distribution over sequences of words. It is trained on a large text corpora. A probability distribution is learned over the proper sequencing of words in a spoken language. The LM is used to enhance the output of the ASR models by correcting its mistakes according to the LM's learned contextual probability of the language. The size of the corpora used to train the LM highly contributes to the accuracy of the model. In this work, we use two models **small** and **large** to refer to the LM corpora size.

In addition to this, a subjective evaluation through listening to a large subset of the anonymized utterances using the GAN-pool revealed significant corruption in the anonymized voice with a bee-like nature. This contradicts with the objective of anonymization in which it is required to produce a human-like voice. Based on this, the use of vanilla GAN for x-vector generation is dropped at this stage and no further experiments using it were conducted.

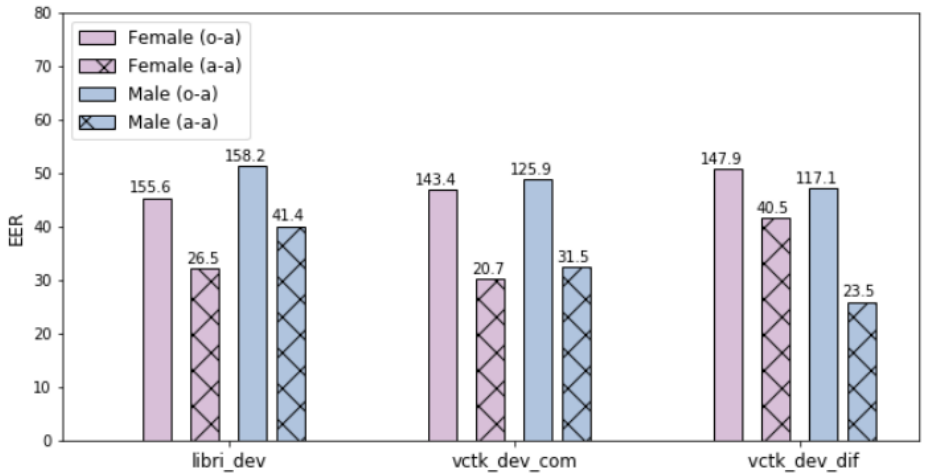

Figure 4.3: GAN-generated Pool: EER (%) Score Obtained by the ASV Evaluation System on Multiple Test Sets. The C_{llr} Score is Displayed on the Top of Each Bar. o–original, a–anonymized Speech Data for Enrollment and Trial Parts.

Table 4.5: GAN-generated pool: ASR results for development data.

#	**Data set**	**Data**	**WER, %**, small	**WER, %, large**
1	libri_dev	a	91.03	90.23
2	vctk_dev	a	91.44	90.53

4.4.2.2 CTGAN Pool Evaluation

In this experiment, the effect of varying the pool size of speaker identities in the anonymization process is studied. Results are reported for Speaker verifiability/linkability metrics and WER for various pool sizes (1000,3000,5000), for both genders, and on multiple development and test datasets.

In Figure 4.4 and Figure 4.5, a steady performance is observed over various pool sizes across multiple datasets. This indicates that the speaker verifiability metrics don't depend on the size of the anonymization pool. It is also noticed that EER and C_{llr} values are lower in the case of (a-a), i.e. both utterances are anonymized. This is a result of the anonymization projecting the utterances to relatively similar speaker identities. In Figure 4.6, it can be noticed that there is minimal difference in WER values due to varying the pool size. It can be noticed that in the case of large LM, WER tends to be better as the larger LM contributes more to correcting the ASR output mistakes. As a result of the varying pool size experiment, we used the smaller pool size throughout the rest of the experiments to shrink the size of the whole anonymization system.

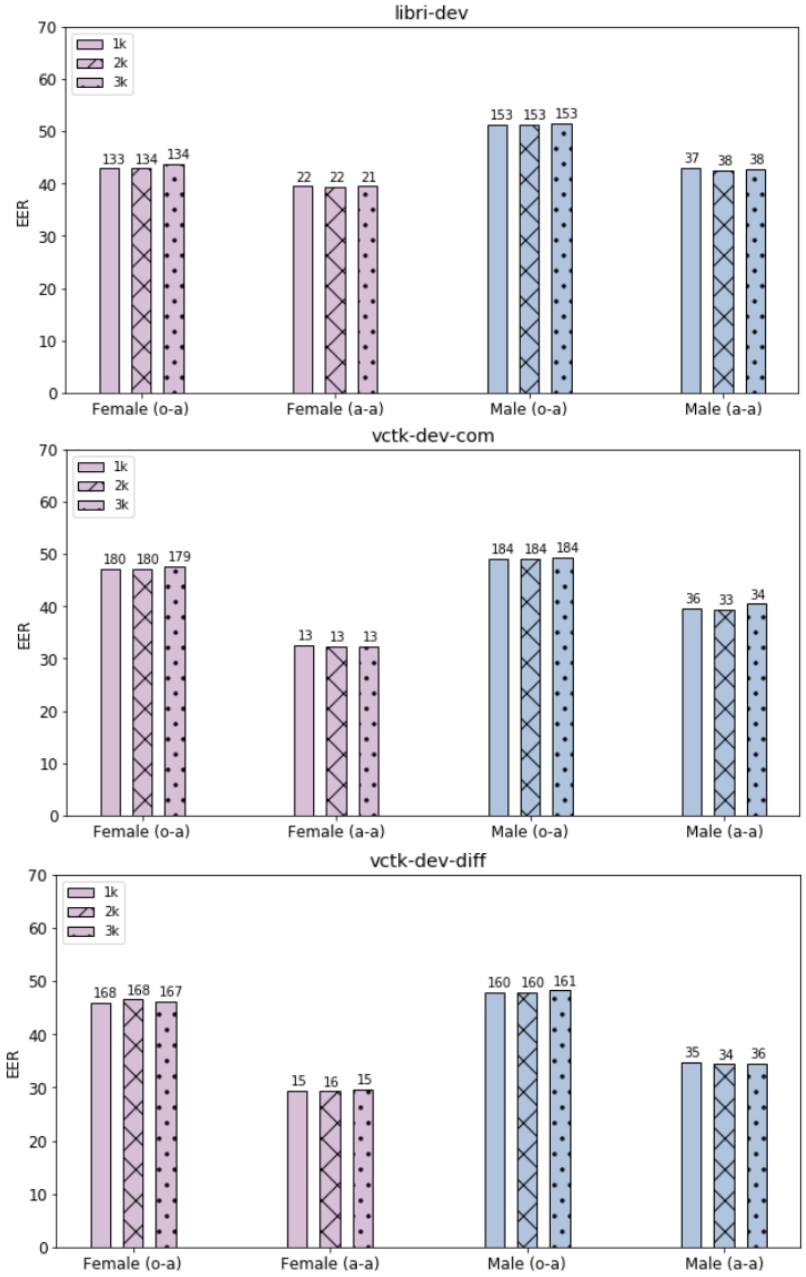

Figure 4.4: ASV Results for CTGAN-generated Pool on Development Sets. The C_{llr} Score is Displayed on the Top of Each Bar.

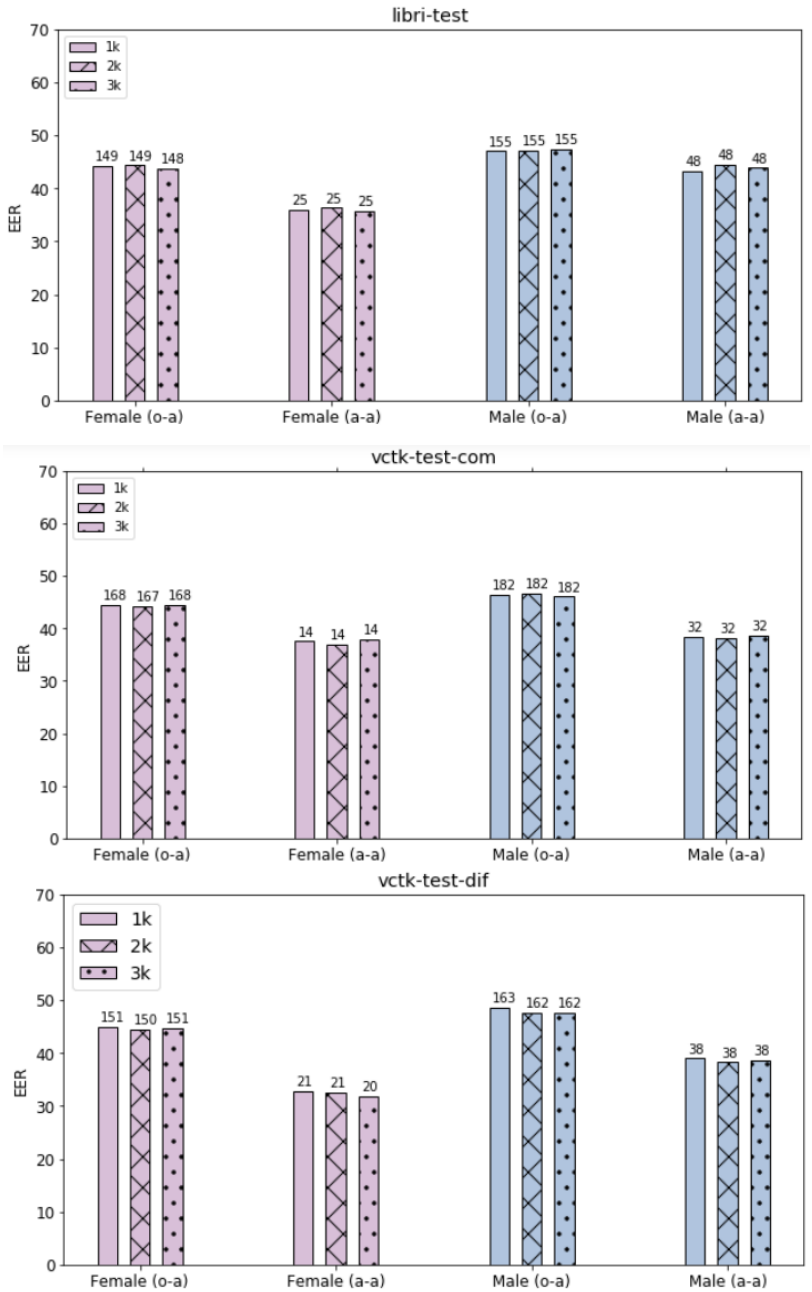

Figure 4.5: ASV Results for CTGAN-generated Pool on Test Sets. The C_{llr} Score is Displayed on the Top of Each Bar.

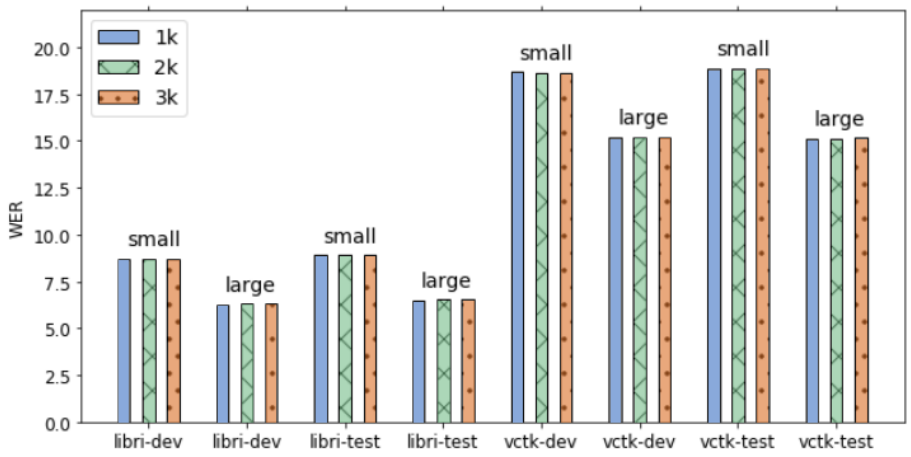

Figure 4.6: ASR results for CTGAN-generated pool with different Pool sizes. Small/Large refer to the language model size.

As discussed earlier, the x-vector selection process from the pool relies on calculating a distance measure, which could be PLDA distance or Cosine. The following experiment compares both distance measures to study the effect on anonymization performance. Results presented in Figure 4.7, Figure 4.8 and Figure 4.9 show comparable performance, where the average of EER values over all datasets for the PLDA case is 42.2, and 43.03 for the cosine case. Whereas the average C_{llr} scores for PLDA is 96.47 as compared to 95.1 for cosine.

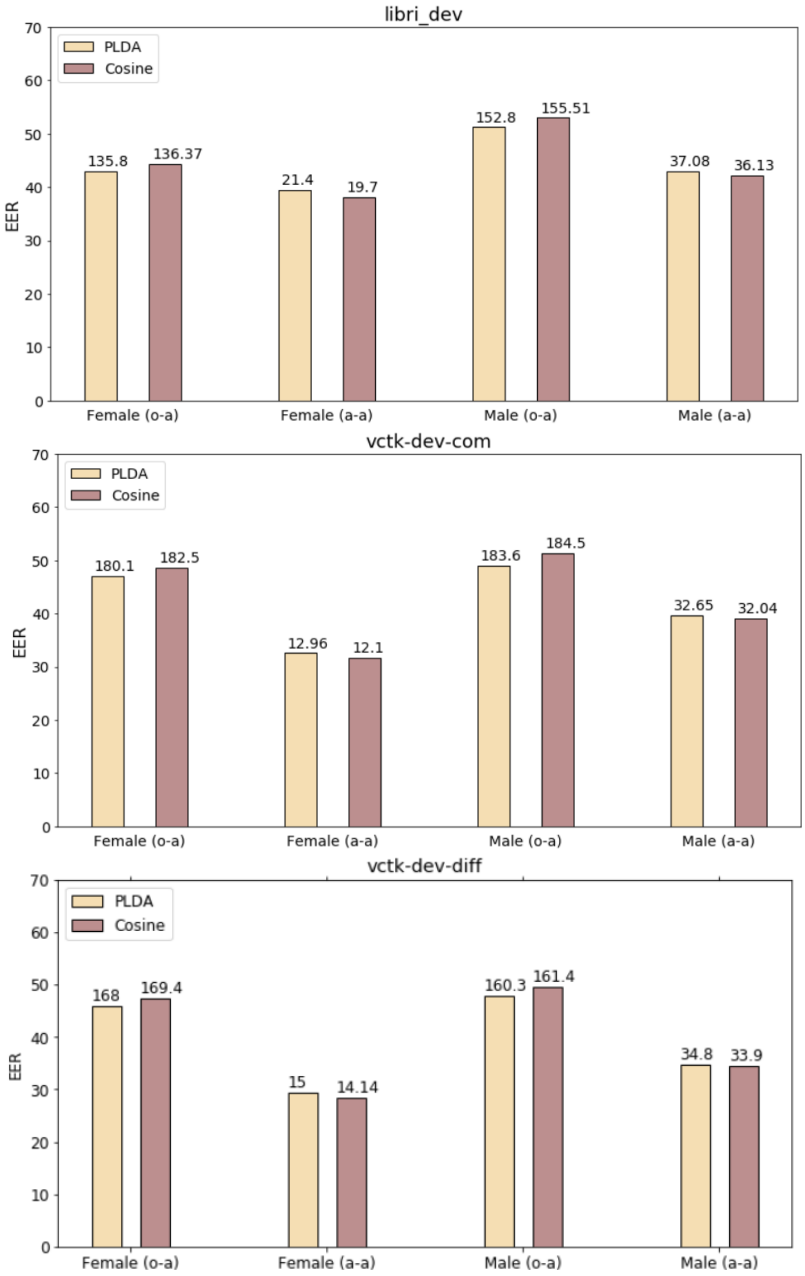

Figure 4.7: ASV Results for Development Partition for PLDA and Cosine Distances. The C_{llr} Score is Displayed on the Top of Each Bar.

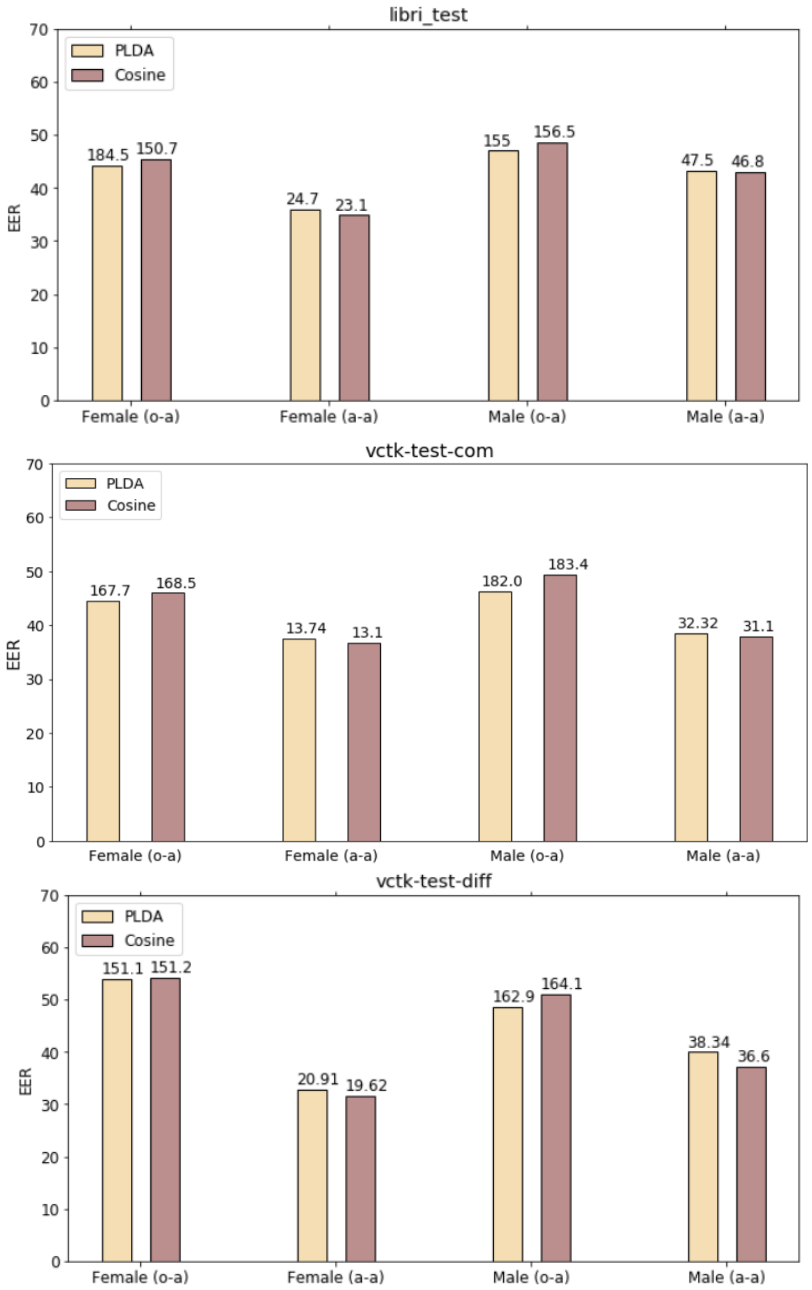

Figure 4.8: ASV Results for Test Partition for PLDA and Cosine Distances. The C_{llr} Score is Displayed on the Top of Each Bar.

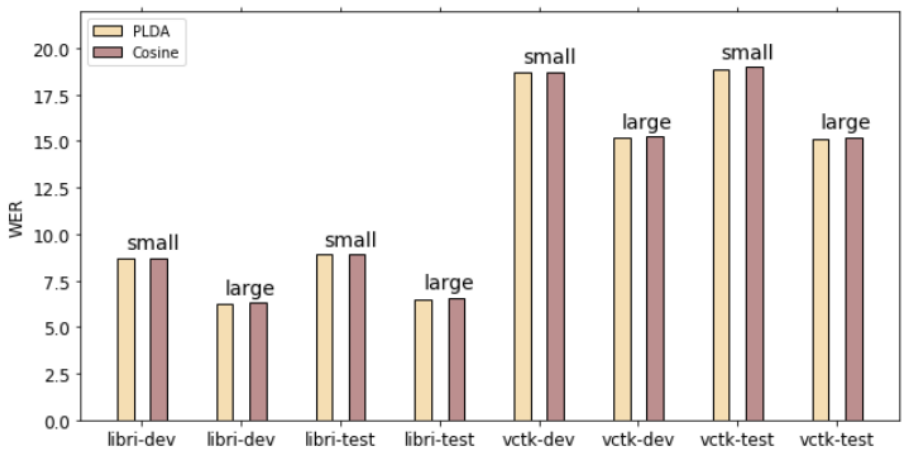

Figure 4.9: ASR results for both development and test data for PLDA and Cosine distances. Small/Large refer to the language model size.

4.4.3 Comparison with Previous Work

In this section, the performance of the proposed anonymization system is compared with the baseline anonymization system [4] and the enhanced system proposed in [69] which uses Gaussian Mixture Models to generate the x-vectors pool.

Figure 4.10 shows the values of the ASV metrics for the three systems. All the alternating values for ASV are within the acceptable range for hiding the speaker identities. A typical ASV system has WER less than 2, indicating that the false acceptance and false rejection rates are below 2, which matches the objective of a general ASV system. However, in the anonymization case, values around 50% indicate that 50% of the times the systems miss-identifies a given speaker as another one as if it's tossing a coin, indicating that the ASV system is actually fooled and the two compared speakers are totally different from each other, which is the exact goal of anonymization. The C_{llr}^{min} represents how well are two classes separated from each other, which is also comparable among the three studies with all values falling within a range consistent with the anonymization objective (above 0.8).

In Figure 4.11, we compare WER and show that our proposed anonymization system achieves state of the art performance. The proposed solution can better preserve the linguistic content of the anonymized speech and make it sound more intelligible. This is a very important aspect in anonymization as these data are still needed for training and personalization purposes, therefore a good speech quality is essential for the data to be still useful.

This comparison can be best viewed in the light of the fact that using synthesized speaker speech identities, we were able to enhance the anonymization process as compared to using real identities.

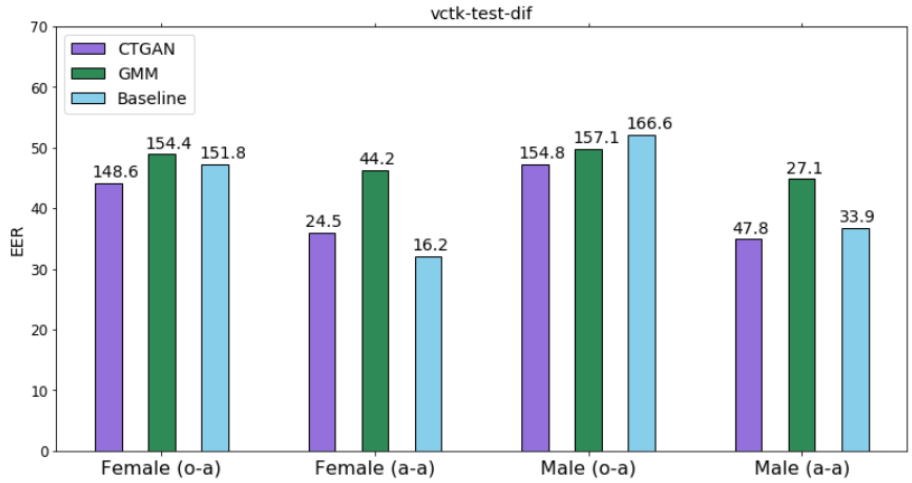

Figure 4.10: ASV Results for Testing Datasets of CTGAN, GMM [69], and Baseline [4].

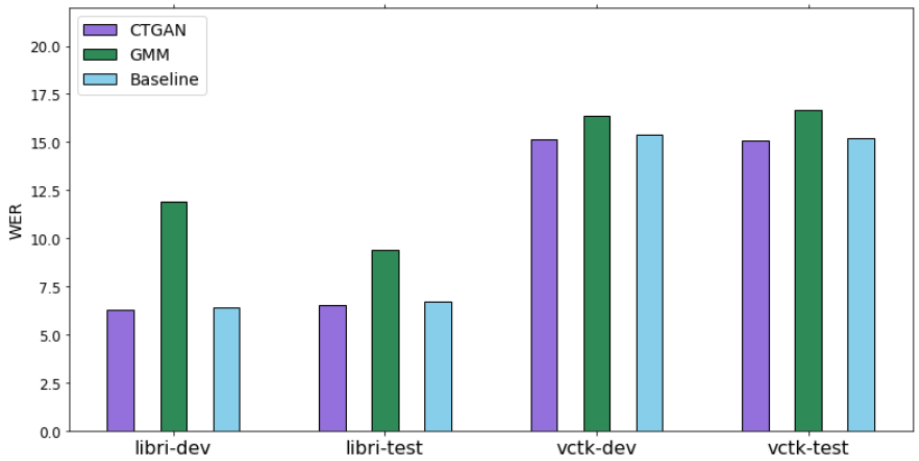

Figure 4.11: ASR Results for Testing Datasets of CTGAN, GMM [69], and Baseline [4].

Chapter 5

Conclusion and Future Work

In this chapter, we summarise the work done in this thesis; highlighting the nature of the problem, suggested solution, and evaluation results. We also discuss some of the possible enhancements and applications as future work.

5.1 Conclusion

Motivated by the recent European privacy legislation, e.g., the general data protection regulation (GDPR), mounting calls for protecting the privacy of speech data have emerged. The problem of speaker anonymization aims at suppressing the personally identifiable information in a speech signal, leaving all other aspects intact.

In this thesis, we examined the problem of speaker anonymization. This work was based on the baseline anonymization system proposed by Interspeech 2020 Challenge: Voice Privacy [4]. We made use of the capabilities of generative adversarial networks to generate authentic speaker identities i.e. x-vectors [48] that could be further utilized in the speaker anonymization process. This chapter summarizes the main findings of our work.

The VoicePrivacy challenge served as an initiative to spread the effort of developing speech privacy-preserving solutions amongst the research community. The objective of the challenge was to develop a system that would output a speech waveform that hides the original speaker's identity while preserving the intelligibility and naturalness of the speech as much as possible.

In that regard, the challenge provided a baseline for achieving such a goal [4] as the first effort to mitigate the presented privacy issues. The work in this thesis was based on the baseline system in [4].

The basic idea of the baseline system is to disentangle the speaker information from the linguistic information in a given spoken utterance. Specifically, three different features are extracted from the input utterance: fundamental frequency (f_o), phoneme posteriogram (PPG), and x-vector. After they replace the x-vector with a pseudo speaker identity from an external pool of speakers, extracted features are merged through a pre-trained synthesizer model that produces mel-spectrograms which will further be converted to a speech waveform using a special vocoder. This system suffers from performance degradation due to the widely averaged speaker identities.

In this thesis, we proposed an anonymization technique in which we trained and evaluated 4 different designs of generative adversarial networks in order to generate authentic and never existing x-vectors and chose the best design which was a CTGAN model. Evaluation metrics involved the study of fake data distribution by calculating the cross-cosine similarity measures, measuring the distance between fake and real distributions in addition to speaker verifiability and intelligibility metrics. Performing speaker anonymization using this approach guarantees the extreme difficulty of reversing the anonymized utterances back to the original speaker by the use of sophisticated attacker systems as the pseudo identities are synthesized artificially, thus adding an extra anonymization layer to the whole process and ensuring better quality for the anonymized utterances, making them more real and close to being natural.

Using x-vectors generated by a CTGAN trained model, a pool of synthesized pseudo speaker identities is created. The anonymization process is followed just like the baseline while achieving better results in terms of WER (6.27% / 6.5% on libri dev/test benchmarks) and comparable results in EER. These results highlight the fact that using artificially synthesized speaker identities achieves better performance than using original x-vectors.

More generally, we believe that this is the first attempt in synthesizing human speaker identities using generative adversarial networks. The scope of use for such a model extends beyond just anonymization. For example, text-to-speech systems and data augmentation for creating speech corpora to train different kinds of models.

5.2 Future Work

As discussed in the previous chapter, the proposed GAN-based anonymization system presented in this thesis showed potential improvement over the baseline system and the GMM-based system [4, 69]. However, analyzing the results and hearing some samples revealed that there is a degradation in quality occurring in cases where the chosen pseudo speaker identity is very far from the original one. As the fundamental frequency and x-vector represent similar information about the identity

of the speaker in terms of voice tone or speech style, combining them when they carry different information corrupts the resulting signal.

As a matter of fact, the study in [70] presented the idea of incorporating a linearly-transformed version of f_o in synthesizing the anonymized speech. We believe that taking this modification one step further to be included in the generative model training will have a positive impact on the anonymization quality and reduce the risk of linking the anonymized utterance back to the original speaker. This would encapsulate the information carried in f_o with that encoded in the x-vector.

Some of the work that can be considered is to study and evaluate the use of the proposed anonymization system in other speech-related systems such as multi-speaker text-to-speech (TTS) where speech data can be synthesized using speaker identities and text inputs, thus perform data augmentation to create data useful for training speech recognition models in any desired low resourced languages.

Bibliography

[1] M. A. Pathak, B. Raj, S. D. Rane, and P. Smaragdis, "Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise," *IEEE signal processing magazine*, vol. 30, no. 2, pp. 62–74, 2013.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, p. 3152676, 2017.

[4] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," *arXiv preprint arXiv:1905.13561*, 2019.

[5] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé, *et al.*, "The voiceprivacy 2020 challenge plan," 2020.

[6] R. Knote, A. Janson, L. Eigenbrod, and M. Söllner, "The what and how of smart personal assistants: principles and application domains for is research," *Multikonferenz Wirtschaftsinformatik (MKWI)*, 2018.

[7] "voice-search-mobile-use-statistics." https://www.thinkwithgoogle.com/marketing-strategies/search/voice-search-mobile-use-statistics/, May 2021.

[8] "Speech to text automated transcription usage worldwide in 2020, by industry and frequency." https://www.statista.com/statistics/1133885/automated-transcription-usage-worldwide-by-industry-and-frequency/, May 2021.

[9] M. Persaud, "Where is voice tech going?." https://techcrunch.com/2020/07/29/voice-tech-in-2020/, July 2020.

[10] "Global speech recognition market." https://www.marketresearchfuture.com/reports/speech-recognition-market-1815, February 2021.

[11] "Captioning and subtitling solution market size, status and forecast 2021-2027." https://reports.valuates.com/market-reports/QYRE-Auto-916/global-captioning-and-subtitling-solution, May 2019.

[12] C. Olson, "New report tackles tough questions on voice and ai." https://about.ads.microsoft.com/en-us/blog/post/april-2019/new-report-tackles-tough-questions-on-voice-and-ai, April 2019.

[13] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *The Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637–642, 1952.

[14] H. F. Olson and H. Belar, "Phonetic typewriter," *The Journal of the Acoustical Society of America*, vol. 28, no. 6, pp. 1072–1081, 1956.

[15] D. B. Fry, "Theoretical aspects of mechanical speech recognition," *Journal of the British Institution of Radio Engineers*, vol. 19, no. 4, pp. 211–218, 1959.

[16] J. W. Forgie and C. D. Forgie, "Results obtained from a vowel recognition computer program," *The Journal of the Acoustical Society of America*, vol. 31, no. 11, pp. 1480–1489, 1959.

[17] T. B. Martin, A. Nelson, and H. Zadell, "Speech recognition by feature-abstraction techniques.," tech. rep., RAYTHEON CO WALTHAM MASS, 1964.

[18] D. K. Dansena and Y. Rathore, "A survey paper on automatic speech recognition by machine," *(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 2918-2922*, 2015.

[19] V. Velichko and N. Zagoruyko, "Automatic recognition of 200 words," *International Journal of Man-Machine Studies*, vol. 2, no. 3, pp. 223–234, 1970.

[20] L. Rabiner, S. Levinson, A. Rosenberg, and J. Wilpon, "Speaker-independent recognition of isolated words using clustering techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 4, pp. 336–349, 1979.

[21] S. Nakagawa, "A survey on automatic speech recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 85, no. 3, pp. 465–486, 2002.

[22] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247. Springer Science & Business Media, 2012.

[23] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16–24, 2010.

[24] V. Radha and C. Vimala, "A review on speech recognition challenges and approaches," *doaj. org*, vol. 2, no. 1, pp. 1–7, 2012.

[25] A. Y. Vadwala, K. A. Suthar, Y. A. Karmakar, and N. Pandya, "Survey paper on different speech recognition algorithm: challenges and techniques," *Int J Comput Appl*, vol. 175, no. 1, pp. 31–36, 2017.

[26] G. Hemakumar and P. Punitha, "Speech recognition technology: a survey on indian languages," *International Journal of Information Science and Intelligent System*, vol. 2, no. 4, pp. 1–38, 2013.

[27] I. Mporas, T. Ganchev, M. Siafarikas, and N. Fakotakis, "Comparison of speech features on the speech recognition task," *Journal of Computer Science*, vol. 3, no. 8, pp. 608–616, 2007.

[28] D. O'Shaughnessy, "Interacting with computers by voice: automatic speech recognition and synthesis," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1272–1305, 2003.

[29] G. Saha, S. Chakroborty, and S. Senapati, "A new silence removal and endpoint detection algorithm for speech and speaker recognition applications," in *Proceedings of the NCC*, vol. 2005, p. 5, Citeseer, 2005.

[30] B. Zamani, A. Akbari, B. Nasersharif, and A. Jalalvand, "Optimized discriminative transformations for speech features based on minimum classification error," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 948–955, 2011.

[31] W. Alkhaldi, W. Fakhr, and N. Hamdy, "Automatic speech/speaker recognition in noisy environments using wavelet transform," in *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, vol. 1, pp. I–463, IEEE, 2002.

[32] V. V. Krishnan and P. B. Anto, "Features of wavelet packet decomposition and discrete wavelet transform for malayalam speech recognition," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, p. 93, 2009.

[33] M. Forsberg, "Why is speech recognition difficult," *Chalmers University of Technology*, 2003.

[34] D. O'Shaughnessy, "Automatic speech recognition: History, methods and challenges," *Pattern Recognition*, vol. 41, no. 10, pp. 2965–2979, 2008.

[35] S. Ranjan, "A discrete wavelet transform based approach to hindi speech recognition," in *2010 international conference on signal acquisition and processing*, pp. 345–348, IEEE, 2010.

[36] X. Tang, "Hybrid hidden markov model and artificial neural network for automatic speech recognition," in *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, pp. 682–685, IEEE, 2009.

[37] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210, IEEE, 2015.

[38] J. Yamagishi, C. Veaux, K. MacDonald, *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[39] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[40] N. Singh, "Automatic speaker recognition: current approaches and progress in last six decades," *Global Journal of Enterprise Information System*, vol. 9, no. 3, pp. 45–52, 2017.

[41] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition .," in *Interspeech*, pp. 1353–1357, 2017.

[42] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[43] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (sitw) speaker recognition database.," in *Interspeech*, pp. 818–822, 2016.

[44] K. Li and E. Wrench, "An approach to text-independent speaker recognition with short utterances," in *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, pp. 555–558, IEEE, 1983.

[45] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[46] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE transactions on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.

[47] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, IEEE, 2014.

[48] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, IEEE, 2018.

[49] E. Conrad, S. Misenar, and J. Feldman, "Chapter 5 - domain 5: Identity and access management (controlling access and managing identity)," in *Eleventh Hour CISSP® (Third Edition)* (E. Conrad, S. Misenar, and J. Feldman, eds.), pp. 117–134, Syngress, third edition ed., 2017.

[50] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[51] C. Olson, "The voice privacy challenge." https://www.voiceprivacychallenge.org/, January 2020.

[52] A. Cohen-Hadria, M. Cartwright, B. McFee, and J. P. Bello, "Voice anonymization in urban sound recordings," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, 2019.

[53] F. Gontier, M. Lagrange, C. Lavandier, and J.-F. Petiot, "Privacy aware acoustic scene synthesis using deep spectral feature inversion," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 886–890, IEEE, 2020.

[54] P. Smaragdis and M. Shashanka, "A framework for secure speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1404–1413, 2007.

[55] S.-X. Zhang, Y. Gong, and D. Yu, "Encrypted speech recognition using deep polynomial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5691–5695, IEEE, 2019.

[56] F. Brasser, T. Frassetto, K. Riedhammer, A.-R. Sadeghi, T. Schneider, and C. Weinert, "Voiceguard: Secure and private speech processing.," in *Interspeech*, vol. 18, pp. 1303–1307, 2018.

[57] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6341–6345, IEEE, 2019.

[58] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?," *arXiv preprint arXiv:2003.14053*, 2020.

[59] K. Hashimoto, J. Yamagishi, and I. Echizen, "Privacy-preserving sound to degrade automatic speaker verification performance," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5500–5504, IEEE, 2016.

[60] J. Qian, H. Du, J. Hou, L. Chen, T. Jung, X.-Y. Li, Y. Wang, and Y. Deng, "Voicemask: Anonymize and sanitize voice input on mobile devices," *arXiv preprint arXiv:1711.11460*, 2017.

[61] Q. Jin, A. R. Toth, T. Schultz, and A. W. Black, "Speaker de-identification via voice transformation," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 529–533, IEEE, 2009.

[62] M. Pobar and I. Ipšić, "Online speaker de-identification using voice transformation," in *2014 37th International convention on information and communication technology, electronics and microelectronics (mipro)*, pp. 1264–1267, IEEE, 2014.

[63] F. Bahmaninezhad, C. Zhang, and J. H. Hansen, "Convolutional neural network based speaker de-identification.," in *Odyssey*, pp. 255–260, 2018.

[64] Y. Han, S. Li, Y. Cao, Q. Ma, and M. Yoshikawa, "Voice-indistinguishability: Protecting voiceprint in privacy-preserving speech data release," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2020.

[65] B. M. L. Srivastava, A. Bellet, M. Tommasi, and E. Vincent, "Privacy-preserving adversarial representation learning in asr: Reality or illusion?," *arXiv preprint arXiv:1911.04913*, 2019.

[66] C. Magarinos, P. Lopez-Otero, L. Docio-Fernandez, E. Rodriguez-Banga, D. Erro, and C. Garcia-Mateo, "Reversible speaker de-identification using pre-trained transformation functions," *Computer Speech & Language*, vol. 46, pp. 36–52, 2017.

[67] T. Justin, V. Štruc, S. Dobrišek, B. Vesnicer, I. Ipšić, and F. Mihelič, "Speaker de-identification using diphone recognition and speech synthesis," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 4, pp. 1–7, IEEE, 2015.

[68] F. Alegre, G. Soldi, and N. Evans, "Evasion and obfuscation in automatic speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 749–753, IEEE, 2014.

[69] H. Turner, G. Lovisotto, and I. Martinovic, "Speaker anonymization with distribution-preserving x-vector generation for the voiceprivacy challenge 2020," *arXiv preprint arXiv:2010.13457*, 2020.

[70] P. Champion, D. Jouvet, and A. Larcher, "A study of f0 modification for x-vector based speech pseudonymization across gender," *arXiv preprint arXiv:2101.08478*, 2021.

[71] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[73] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *2009 IEEE 12th international conference on computer vision*, pp. 2146–2153, IEEE, 2009.

[74] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, JMLR Workshop and Conference Proceedings, 2011.

[75] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *International conference on machine learning*, pp. 1319–1327, PMLR, 2013.

[76] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[77] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[78] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *arXiv preprint arXiv:1606.03498*, 2016.

[79] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[80] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[81] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[82] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *arXiv preprint arXiv:1907.00503*, 2019.

[83] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, 2011.

[84] N. Brümmer and J. Du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[85] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[86] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[87] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[88] X. Wang and J. Yamagishi, "Neural harmonic-plus-noise waveform model with trainable maximum voice frequency for text-to-speech synthesis," *arXiv preprint arXiv:1908.10256*, 2019.

Appendix A

Tabulated Results

Table A.1: GAN-generated Pool: ASV Results for Development Data.

#	**Data set**	**Enroll**	**Trials**	**Gender**	**EER, %**	$\mathbf{C}_{llr}^{min}$	$\mathbf{C}_{llr}$
1	libri_dev	o	a	f	45.310	0.992	155.598
2	libri_dev	a	a	f	32.100	0.881	26.518
3	libri_dev	o	a	m	51.240	0.994	158.181
4	libri_dev	a	a	m	40.060	0.948	41.380
5	vctk_dev_com	o	a	f	46.800	0.985	143.445
6	vctk_dev_com	a	a	f	30.230	0.829	20.699
7	vctk_dev_com	o	a	m	48.720	0.972	125.887
8	vctk_dev_com	a	a	m	32.480	0.841	31.461
9	vctk_dev_dif	o	a	f	50.810	0.986	147.962
10	vctk_dev_dif	a	a	f	41.660	0.969	40.515
11	vctk_dev_dif	o	a	m	47.100	0.990	117.038
12	vctk_dev_dif	a	a	m	25.910	0.696	23.492

Table A.2: ASV Results for CTGAN-generated Pool.

Data	E	T	G	EER, %			C_{llr}^{min}			C_{llr}		
				Pool Size								
				1k	3k	5k	1k	3k	5k	1k	3k	5k
libri-dev	o	a	f	42.9	42.9	43.75	0.96	0.96	0.96	133.3	134.1	133.9
	a	a		39.49	39.35	39.49	0.93	0.92	0.92	21.40	21.43	21.19
	o	a	m	51.24	51.24	51.55	0.98	0.98	0.98	152.8	153.2	153.2
	a	a		43.01	42.55	42.80	0.952	0.952	0.95	37.08	37.43	37.5
libri-test	o	a	f	44.16	44.34	43.80	0.98	0.98	0.98	148.5	149.0	148.3
	a	a		35.95	36.31	35.77	0.89	0.89	0.89	24.72	24.58	24.37
	o	a	m	46.99	47.22	47.44	0.99	0.99	0.99	155.0	154.78	154.8
	a	a		43.21	44.54	43.88	0.97	0.97	0.97	47.56	47.72	48.21
vctk-dev-com	o	a	f	47.09	47.09	47.67	0.973	0.97	0.97	180.1	179.6	178.7
	a	a		32.56	32.27	32.27	0.821	0.82	0.817	12.96	12.73	12.72
	o	a	m	49.00	49.00	49.29	0.99	0.99	0.99	183.6	183.5	184.3
	a	a		39.60	39.32	40.46	0.92	0.92	0.92	32.65	32.40	33.62
vctk-dev-diff	o	a	f	45.93	46.55	46.10	0.95	0.96	0.95	168.0	168.1	167.1
	a	a		29.48	29.25	29.59	0.80	0.799	0.80	15.00	15.53	14.81
	o	a	m	47.79	47.84	48.24	0.99	0.99	0.99	160.30	160.1	161.0
	a	a		34.79	34.39	34.54	0.88	0.88	0.89	34.87	34.38	35.53
vctk-test-com	o	a	f	44.51	44.22	44.51	0.975	0.974	0.975	167.7	167.0	167.5
	a	a		37.57	36.99	37.86	0.878	0.881	0.876	13.74	13.76	13.60
	o	a	m	46.33	46.61	46.05	0.93	0.99	0.99	182.0	181.9	182.03
	a	a		38.42	38.14	38.70	0.94	0.93	0.93	32.32	31.72	32.27
vctk-test-dif	o	a	f	45.01	44.39	44.65	0.99	0.98	0.98	151.1	150.4	150.9
	a	a		32.82	32.61	31.89	0.86	0.851	0.853	20.91	20.81	19.68
	o	a	m	48.56	47.65	47.7	0.998	0.997	0.997	162.9	162.2	162.3
	a	a		38.98	38.29	38.63	0.947	0.943	0.94	38.34	37.50	38.24

Table A.3: ASR Results for CTGAN-generated Pool with Diffrent Pool Sizes.

#	**Data set**	**Data**	**WER, %**, small			**WER, %, large**		
			Pool Size					
			1k	**3k**	**5k**	**1k**	**3k**	**5k**
1	libri_dev	a	8.73	8.73	8.73	6.27	6.32	6.32
2	libri_test	a	8.89	8.94	8.93	6.50	6.57	6.57
3	vctk_dev	a	18.71	18.65	18.65	15.17	15.21	15.21
4	vctk_test	a	18.82	18.83	18.88	15.10	15.09	15.20

Table A.4: ASV Results for both Development and Test Partitions for PLDA and Cosine Distances.

Data	E	T	G	EER, %		C_{llr}^{min}		C_{llr}	
				Distance Choice					
				PLDA	Cosine	PLDA	Cosine	PLDA	Cosine
libri-dev	o	a	f	42.9	44.32	0.96	0.976	135.8	136.37
	a	a		39.49	38.07	0.93	0.916	21.40	19.70
	o	a	m	51.24	52.95	0.98	0.989	152.8	155.51
	a	a		43.01	42.24	0.952	0.944	37.08	36.13
libri-test	o	a	f	44.16	45.44	0.98	0.987	148.5	150.74
	a	a		35.95	34.85	0.89	0.876	24.72	23.06
	o	a	m	46.99	48.55	0.99	0.995	155.0	156.5
	a	a		43.21	42.98	0.97	0.969	47.56	46.808
vctk-dev-com	o	a	f	47.09	48.55	0.973	0.979	180.1	182.5
	a	a		32.56	31.69	0.821	0.81	12.96	12.1
	o	a	m	49.00	51.28	0.99	0.995	183.6	184.5
	a	a		39.60	39.03	0.92	0.91	32.65	32.04
vctk-dev-diff	o	a	f	45.93	47.33	0.95	0.96	168.0	169.4
	a	a		29.48	28.35	0.80	0.78	15.00	14.14
	o	a	m	47.79	49.63	0.99	1.00	160.30	161.4
	a	a		34.79	34.5	0.88	0.88	34.87	33.9
vctk-test-com	o	a	f	44.51	45.95	0.975	0.9	167.7	168.5
	a	a		37.57	36.71	0.878	0.86	13.74	13.1
	o	a	m	46.33	49.44	0.93	0.99	182.0	183.4
	a	a		38.42	37.85	0.94	0.92	32.32	31.1
vctk-test-dif	o	a	f	45.01	45.27	0.99	0.991	151.1	151.2
	a	a		32.82	31.5	0.86	0.84	20.91	19.62
	o	a	m	48.56	51.09	0.998	1.0	162.9	164.1
	a	a		38.98	37.2	0.947	0.93	38.34	36.6

Table A.5: ASR Results for both Development and Test Data for PLDA and Cosine Distances.

#	Data set	Data	WER, %, small		WER, %, large	
			Distance Choice			
			PLDA	Cosine	PLDA	Cosine
1	libri_dev	a	8.73	8.69	6.27	6.34
2	libri_test	a	8.89	8.95	6.50	6.54
3	vctk_dev	a	18.71	18.70	15.17	15.24
4	vctk_test	a	18.82	18.974	15.10	15.21

Table A.6: ASV Results for Testing Datasets of CTGAN (C), GMM (G) [69], and Baseline (B) [4].

Data	E	T	G	EER, %			C_{llr}^{min}			C_{llr}		
				Pool Origin								
				C	G	B	C	G	B	C	G	B
Libri_Test	o	a	f	44.1	48.9	47.2	0.98	0.99	0.99	148.6	154.4	151.8
	a	a		36.01	46.2	32.1	0.89	0.99	0.83	24.5	44.2	16.2
	o	a	m	47.2	49.7	52.1	0.99	0.97	0.99	154.8	157.1	166.6
	a	a		43.87	44.8	36.75	0.97	0.96	0.90	47.8	27.1	33.9
VCTK_Test	o	a	f	44.6	45.1	48.0	0.98	0.99	0.99	150.8	147.2	146.9
	a	a		32.4	42.1	31.7	0.85	0.96	0.84	20.4	16.3	11.5
	o	a	m	47.9	48.8	53.8	0.99	1	1	162.5	158.4	1.67
	a	a		38.63	49.5	30.9	0.94	0.96	0.83	38.0	30.7	23.8

Table A.7: ASR (WER) Results for Development and Test Data of CTGAN (C), GMM (G) [69] and Baseline (B) [4].

#	Data set	Data	C	G	B
1	libri_dev	a	6.27	11.89	6.39
2	libri_test	a	6.50	9.38	6.73
3	vctk_dev	a	15.17	16.35	15.38
4	vctk_test	a	15.10	16.65	15.23

إخفاء هوية المتحدّث باستخدام شبكات الخصومة التوليديّة

إعداد

آية سليمان الجعفري

المشرف

أمجد الموسى

المشرف المساعد

إياد جعفر

الملخص

أتاح استخدام الأجهزة الذكية إنتاج العديد من التطبيقات التي تمكّن المستخدم من التفاعل معها بأشكال مختلفة. و يعتبر الكلام الشكل الطبيعي والأكثر شيوعًا لتفاعل المستخدمين مع أجهزتهم الذكيّة. إلا أن المادة الصوتيّة تشكّل مصدراً وفيراً للمعلومات الشخصيّة الحسّاسة التي تخص المستخدم، وبالتالي فإن احتماليّة تسرّب الكلام الخاص بالمستخدم تشكل خطراً على خصوصيته وتهدّد حريته في التعبير. ويكمن خطر تسريب المادة الصوتية في امكانية فصل الهوية الصوتية (النّبرة) ومن ثم إعادة تركيبها مع تشكيلة كلمات أخرى قد يتم استخدامها لاختراق أنظمة التحقق الصوتية الخاصة بالمستخدم وبالتالي اختراق خصوصيته.

يتمثل أحد الحلول لهذه المشكلة في معالجة المادة الصوتية للمستخدم بحيث يتم إخفاء هوية المتحدث الصوتية واستبدالها بهوية ثانية قبل مشاركتها. في هذه الأطروحة، يتم استخدام هويات صوتية مزيفة لأصوات بشرية لإخفاء هوية المتحدث الرئيسي. اعتمد هذا الحل على استخدام شبكات الخصومة التوليدية لإنتاج هويات صوتية بشريّة تعمل على تحسين عملية إخفاء هوية المتكلّم.

تمت تجربة العديد من أنواع شبكات الخصومة التوليدية من أجل تحقيق أفضل أداء في إخفاء هوية المتحدث. وقد حققت شبكة الخصومة التوليدية الجدولية الشرطية أفضل أداء بالاعتماد

على عدّة معايير تقييم استُخدمت للحكم على جودة الهويات الصوتية المُنتَجَة من قِبَل الشبكة. كما أثبتت النتائج تفوق أداء نهج إخفاء الهوية المقترح على أفضـــل أنظمة إخفاء الهوية المتاحة من حيث القدرة على إنتاج كمية متنوعة من الهويات الصــوتية للمتحدثين (تم تقييمها باســتخدام توزيع تشــابه جيب التمام) ونســبة التقارب بين الهويات المزيفة والحقيقية (تم تقييمها باســتخدام اختبار كولموغوروف–ســميرنوف). بالإضـــافة إلى ذلك، فإن معدّل الكلمات الخاطئة التي حققها نظام خارجي للتعرّف التلقائي على الكلام كانت قيمته 6.27% و 6.5% على مجموعتي البيانات المعيارية libri–dev و libri–test على التوالي.

الكلمات المفتاحية: إخفاء هوية المتحدث، خصوصية الصوت، شبكات الخصومة التوليدية، شبكة الخصومة التوليدية الجدولية الشرطية، هوية الصوت.