Speaker Anonymization: Representation, Evaluation and Formal Guarantees

Brij Mohan Lal Srivastava

École Doctorale Mathematiques, Sciences du Numerique et de leurs Interactions (MADIS)

Thèse de Doctorat

Anonymisation du Locuteur

Représentation, Évaluation et Garanties Formelles

préparée et soutenue publiquement par

Brij Mohan Lal Srivastava

à **Villeneuve d'Ascq** le **2 décembre 2021**, pour obtenir le grade de

Docteur en Informatique

Encadrants: Dr. Aurélien Bellet
Dr. Emmanuel Vincent
Prof. Marc Tommasi

Soutenue devant le jury composé de:

Sonia Ben Mokhtar	CNRS - LIRIS	Examinatrice
Hamed Haddadi	Imperial College London	Rapporteur
Sylvain Meignier	Le Mans Université	Rapporteur & Président du jury
Aurélien Bellet	Inria Lille – Nord Europe	Co-encadrant
Emmanuel Vincent	Inria Nancy – Grand Est	Co-encadrant
Marc Tommasi	Université de Lille	Directeur

Doctoral school Mathematics and Digital Sciences (MADIS)

Doctoral Dissertation

Speaker Anonymization

Representation, Evaluation and Formal Guarantees

prepared and publicly defended by

Brij Mohan Lal Srivastava

in **Villeneuve d'Ascq** on **December 2nd, 2021**, to obtain the degree of

Doctor in Computer Science

Supervisors: Dr. Aurélien Bellet
Dr. Emmanuel Vincent
Prof. Marc Tommasi

Defended before the jury composed of:

Sonia Ben Mokhtar	CNRS - LIRIS	Examiner
Hamed Haddadi	Imperial College London	Reviewer
Sylvain Meignier	Le Mans Université	Reviewer & President of the jury
Aurélien Bellet	Inria Lille – Nord Europe	Co-supervisor
Emmanuel Vincent	Inria Nancy – Grand Est	Co-supervisor
Marc Tommasi	Université de Lille	Supervisor

Dedicated to my loving family...

Acknowledgements

With great reverence, I would like to acknowledge the role of my three supervisors, Aurélien, Emmanuel, and Marc, without whom this body of research would not be possible. Although it is hard for me to describe in words the gratefulness that I feel towards them, I will make an effort to carry forward this tradition. First and foremost, with an inspiring level of synchronization and bonhomie among themselves, they work as a power-packed unit which constantly boosted my motivation to push forth and make progress in this new, multidisciplinary field of speech privacy. They patiently listened to me every week, corrected me every time I made mistakes, and encouraged me to choose the most relevant directions through constructive debates. Shielding me from the qualms of moving into a new country, working from two different cities, and everything else that could have broken my momentum, they also trained me to be patient, to persevere, and lead good scientific efforts. Moreover, at every step, they lead by example to demonstrate how a researcher and a mentor must proceed to achieve his/her goals with never-dying motivation, humility, and curiosity. There is no question about their unwavering dedication towards accurate science and ethical research practices, which is also ingrained in Inria's work culture. Such a culture promotes socially-aware scientific progress and raises intriguing, authentic questions that are the lifeblood of research.

Working at Inria, with an extremely supportive and intellectual cohort, is a landmark experience. I am fortunate to have such friendly and respectful team members from both Magnet at Lille and Multispeech at Nancy. My friends Arijus, Carlos, Cesar, Mahsa, Mariana, Mathieu, Onkar, and William have been my stress busters and a company for every day, without whom I cannot imagine finishing my thesis. We shared beautiful times and supported each other in good and bad moments. Despite covid-19 confinements imposed during my one year in Nancy, I developed deep friendships with my colleagues Arash, Ashwin, Manu, Nico, Sahid, Shakeel, Sunit, and Tulika. Some of us explored the city together, held long technical and sometimes political discussions over tea, cooked together, and even collaborated in research projects.

I would also like to especially acknowledge the support of Nathalie and Mohamed, who have actively contributed their efforts towards the research done in this thesis. Since the start of my PhD, Nathalie shared the workload by implementing some of the research ideas and contributing to the codebase of many parts related to the H2020 COMPRISE project. She streamlined the research prototypes developed during this thesis using best engineering practices which made experimentation a lot quicker. Mohamed also helped significantly to progress the research in this thesis. Even though the field of speech processing was new to him, he quickly learned the necessary skills and produced excellent research in the field of speech privacy. Because of his mathematical dexterity, he has been an excellent collaborator and helped to simplify several computations. It was a pleasure working with both of them and I wish we collaborate more in the future.

This section remains incomplete without the mention of my family who provided their unconditional support for me to pursue my interest, even though it has been gloomy at times to be away from them. My parents, my wife, my sisters, and my in-laws showered me with love and constantly checked on my mental and physical well-being. I am grateful for the fact that they always took good care of themselves which left me without worries to focus on my research.

I would like to gratefully acknowledge the financial support provided by Inria, the H2020 COMPRISE project, and the University of Lille which was helpful to sustain myself. Last but not the least, I would like to express my gratitude towards the machines of Magnet, Grid5000[1] and the infrastructure support team, which kindly let me use their GPUs for accelerating my experiments, and gave me special access in times of urgent need.

[1] https://www.grid5000.fr/

Abstract

Large-scale centralized storage of speech data poses severe privacy threats to the speakers. Indeed, the emergence and widespread usage of voice interfaces starting from telephone to mobile applications, and now digital assistants have enabled easier communication between the customers and the service providers. Massive speech data collection allows its users, for instance researchers, to develop tools for human convenience, like voice passwords for banking, personalized smart speakers, etc. However, centralized storage is vulnerable to cybersecurity threats which, when combined with advanced speech technologies like voice cloning, speaker recognition, and spoofing, may endow a malicious entity with the capability to re-identify speakers and breach their privacy by gaining access to their sensitive biometric characteristics, emotional states, personality attributes, pathological conditions, etc. Individuals and the members of civil society worldwide, and especially in Europe, are getting aware of this threat. With firm backing by the GDPR, several initiatives are being launched, including the publication of white papers and guidelines, to spread mass awareness and to regulate voice data so that the citizens' privacy is protected.

This thesis is a timely effort to bolster such initiatives and propose solutions to remove the biometric identity of speakers from speech signals, thereby rendering them useless for re-identifying the speakers who spoke them. Besides the goal of protecting the speaker's identity from malicious access, this thesis aims to explore the solutions which do so without degrading the usefulness of speech. We present several anonymization schemes based on voice conversion methods to achieve this two-fold objective. The output of such schemes is a high-quality speech signal that is usable for publication and a variety of downstream tasks. All the schemes are subjected to a rigorous evaluation protocol which is one of the major contributions of this thesis. This protocol led to the finding that the previous approaches do not effectively protect the privacy and thereby directly inspired the VoicePrivacy initiative which is an effort to gather individuals, industry, and the scientific community to participate in building a robust anonymization scheme. We introduce a range of anonymization schemes under the purview of the VoicePrivacy initiative and empirically prove their superiority in terms of privacy protection and utility. Finally, we endeavor to remove the residual speaker identity from the anonymized speech signal using the techniques inspired by differential privacy. Such techniques provide provable analytical guarantees to the proposed anonymization schemes and open up promising perspectives for future research.

In practice, the tools developed in this thesis are an essential component to build trust in any software ecosystem where voice data is stored, transmitted, processed, or published. They aim to help the organizations to comply with the rules mandated by civil governments and give a choice to individuals who wish to exercise their right to privacy.

Résumé des travaux en français

L'émergence et la généralisation des interfaces vocales présentes dans les téléphones, les applications mobiles et les assistants numériques ont permis de faciliter la communication entre les citoyens, utilisateurs d'un service, et les prestataires de services. Citons à titre d'exemple l'utilisation de mots de passe vocaux pour les opérations bancaires, des haut-parleurs intelligents personnalisés, etc. Pour réaliser ces innovations, la collecte massive de données vocales est essentielle aux entreprises comme aux chercheurs. Mais le stockage centralisé à grande échelle des données vocales pose de graves menaces à la vie privée des locuteurs. En effet, le stockage centralisé est vulnérable aux menaces de cybersécurité qui, lorsqu'elles sont combinées avec des technologies vocales avancées telles que le clonage vocal, la reconnaissance du locuteur et l'usurpation d'identité peuvent conférer à une entité malveillante la capacité de ré-identifier les locuteurs et de violer leur vie privée en accédant à leurs caractéristiques biométriques sensibles, leurs états émotionnels, leurs attributs de personnalité, leurs conditions pathologiques, etc. Les individus et les membres de la société civile du monde entier, et particulièrement en Europe, prennent conscience de cette menace. Avec l'entrée en vigueur du règlement général sur la protection des données (RGPD), plusieurs initiatives sont lancées, notamment la publication de livres blancs et de lignes directrices, pour sensibiliser les masses et réguler les données vocales afin que la vie privée des citoyens soit protégée.

Cette thèse constitue un effort pour soutenir de telles initiatives et propose des solutions pour supprimer l'identité biométrique des locuteurs des signaux de parole, les rendant ainsi inutiles pour ré-identifier les locuteurs qui les ont prononcés. Outre l'objectif de protéger l'identité du locuteur contre les accès malveillants, cette thèse vise à explorer les solutions qui le font sans dégrader l'utilité de la parole. Nous présentons plusieurs schémas d'anonymisation basés sur des méthodes de conversion vocale pour atteindre ce double objectif. La sortie de tels schémas est un signal vocal de haute qualité qui est utilisable pour la publication et pour un ensemble de tâches en aval. Tous les schémas sont soumis à un protocole d'évaluation rigoureux qui est l'un des apports majeurs de cette thèse. Ce protocole a conduit à la découverte que les approches existantes ne protègent pas efficacement la vie privée et a ainsi directement inspiré l'initiative VoicePrivacy qui rassemble les individus, l'industrie et la communauté scientifique pour participer à la construction d'un schéma d'anonymisation robuste. Nous introduisons une gamme de schémas d'anonymisation dans le cadre de l'initiative VoicePrivacy et prouvons empiriquement leur supériorité en termes de protection de la vie privée et d'utilité. Enfin, nous nous efforçons de supprimer l'identité résiduelle du locuteur du signal de parole anonymisé en utilisant les techniques inspirées de la confidentialité différentielle. De telles techniques fournissent des garanties analytiques démontrables aux schémas d'anonymisation proposés et ouvrent des portes pour de futures recherches.

En pratique, les outils développés dans cette thèse sont un élément essentiel pour établir la confiance dans tout écosystème logiciel où les données vocales sont stockées, transmises, traitées ou publiées. Ils visent à aider les organisations à se conformer aux règles mandatées par les gouvernements et à donner le choix aux individus qui souhaitent exercer leur droit à la vie privée.

Table of contents

List of figures

List of tables

List of acronyms

ACC Accuracy

AM Acoustic Model

ANN Artificial Neural Network

ASI Automatic Speaker Identification

ASR Automatic Speech Recognition

ASV Automatic Speaker Verification

BLSTM Bidirectional Long Short-Term Memory

BN Bottleneck

CE Cross Entropy

CNN Convolutional Neural Network

CTC Connectionist Temporal Classification

dB Decibels

DNN Deep Neural Network

DP Differential Privacy

EER Equal Error Rate

FAR False Acceptance Rate

FRR False Rejection Rate

GDPR General Data Protection Regulation

GPU Graphical Processing Unit

HMM Hidden Markov Model

LDP Local Differential Privacy

LF-MMI Lattice Free-Maximum Mutual Information

LSTM Long Short-Term Memory

MFCC Mel Frequency Cepstral Coefficient

MOS Mean Opinion Score

MSE Mean Squared Error

NCCF Normalized Cross Correlation Function

NSF Neural Source-Filter

PLDA Probabilistic Linear Discriminant Analysis

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent

SPSS Statistical Parametric Speech Synthesis

SS Speech Synthesis

STFT Short-Time Fourier Transform

TDNN-F Factorized Time Delay Neural Network

TDNN Time Delay Neural Network

TTS Text-to-Speech

VAD Voice Activity Detection

VC Voice Conversion

VTLN Vocal Tract Length Normalization

WER Word Error Rate

YAAPT Yet Another Algorithm for Pitch Tracking

Chapter 1

Introduction

> Civilization is the progress toward a society of privacy.
>
> *Ayn Rand*

1.1 Motivation

Speaking and listening are the most convenient, non-tactile and expressive forms of human communication. During oral conversations, we transmit not only the linguistic content, but also paralinguistic and extra-linguistic cues such as our emotional state, age, gender, personality, health state, etc. to our interlocutors [288]. Hence the rich nature of verbal dialog makes it a natural choice as the interface between humans and machines. For over two centuries, researchers have been intrigued by the process of speech generation by humans and machines [140]. The earliest known attempt of producing human-like sounds from a mechanical model was made in the later part of the 18th century by the Russian scientist Christian Kratzenstein [161] and soon after by an Austrian inventor named Wolfgang von Kempelen [310] who is famous for his "speaking machine".

Fast forward to the 1980s, Artificial Neural Networks were successfully introduced to recognize a few words in a speech signal [181]. The field of speech processing has come a long way since then as four decades later and with several groundbreaking advances, Deep Neural Networks (DNNs) have become the state-of-the-art [103, 109] for large vocabulary continuous speech recognition and several associated tasks, such as text-to-speech, speaker recognition, etc. These gigantic networks with several millions of parameters have surpassed human level performance in speech recognition [9], but there are new and extraordinary challenges with the emergence of speech interfaces in the marketplace.

Several smart digital assistants are available in the market today which are powered by the decades of advances in speech recognition and conversational models. The goal of their manufacturers is to make conversations between humans and digital assistants as seamless as possible.[1] They are designed to handle a wide range of commands and questions in a jestful manner, and are usually provided with a name and gender so that users can personify them. Such realizations can build trust between the digital assistant and the human, which helps to enhance the engagement [185]. Users can now control their home appliances, play music, request a joke, shop online and of course send messages among several other functions using the digital assistant. The users of digital assistants are growing at the rate of about 35% per year. Till early

[1] https://developer.amazon.com/alexaprize/about

2019, Amazon had already sold 100 million devices worldwide [29] with Alexa[2], a cloud-based voice-enabled virtual assistant, and the projected market of digital assistant by the end of 2021 is expected to reach 843 million users worldwide, which amounts to a revenue of $15.8 billion [237].

Unfortunately, the most determining factor in the success of the advanced statistical models running the digital assistant ecosystem is the enormous size of their training data sets [127], closely followed by the availability of high computing infrastructure such as Graphical Processing Units (GPUs) [117]. With the availability of pervasive Internet and smart devices, large quantities of speech data is being collected by digital assistant manufacturers like Google, Amazon, Apple and Microsoft. This data is stored at centrally located servers and, depending on the needs, it can be made available to developers, annotators and managers. Among the consumers who own a digital assistant, 65% claim that they do not know everything the device can do [232] and, among those who do not own a digital assistant, only 16% cite privacy reasons not to purchase one. This lack of awareness further opens the doors to a massive privacy breach.

Privacy is considered as a fundamental human right in many regions of the world [19]. It is intimately linked with human dignity and freedom of thought and expression. The Indian Constitution lists the "right to privacy" under Article 21 which deals with protection of life and personal liberty [38]. Yet there is no universally accepted definition of privacy [186] which makes it hard to enforce it as a legally protected right. The extent of technological intrusion into people's lives as described previously poses a severe threat to individual privacy but there is little consensus between technological and legal communities to legislate strong laws for data protection. In 1890 Warren and Brandeis [317] defined privacy for the first time as the "general right of the individual to be let alone". Since then the European Union and several countries such as the United States of America and Canada [174] have included some privacy articles in their constitution.

In 2016, the European Union passed the General Data Protection Regulation (GDPR) [77] setting a historical precedent for data privacy law worldwide. The GDPR is listed under the EU Charter of Fundamental Rights which stipulates that European citizens have right to protection of their personal data.[3] The law clearly holds companies accountable for users' data, users have complete control over the usage and distribution of their data and can request deletion at any time they want. Moreover, the law is applicable over the data of citizens of all the member states even if it was processed overseas. The violators are heavily fined [13] if found guilty, causing companies to block users from Europe [159] to avoid non-compliance issues. Although Section 2 of the GDPR mentions guidelines to ensure the "security of personal data" through pseudonymisation and encryption, there is a clear lack of understanding with respect to the capture, storage and processing of speech data. Recently, Nautsch et al. [213] launched a collaborative effort to harmonise the terminology between speech researchers and legal experts so that the sensitive attributes in speech signals could be clearly understood by the legislators.

Recently the French Data Protection Authority (CNIL) published a white paper [46] to explore legal, technical and ethical issues associated with voice assistants. The paper briefly mentions the work done during this thesis as the potential solution to some of the technical issues. At the time of writing this thesis, the European Data Protection Board (EDPB) also released guidelines [78] on virtual voice assistants for different stakeholders involved in their production and use. Although the guidelines focus on the legal bases which empower digital assistant users to request data erasure and voice sanitisation techniques to remove situational information and background noise, they are not very encouraging of voice anonymization methods due to several open challenges that impede the evolution of the technology. Nevertheless, the guidelines were open for public consultation calling for further technological as well as legal development in this direction to explore exact grounds for processing speech data and clear jurisdiction upon violation.

[2] https://developer.amazon.com/en-US/alexa

[3] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en

Speech is a biometric characteristic of human beings [137], which can produce distinguishing and repeatable biometric features. According to Article 4(14) of the GDPR, voice data is inherently biometric personal data which relates to physical, physiological or behavioural characteristics of a natural person. Voiceprints [240], which are used to identify speakers, are also deployed in payment services [23] for authenticating transactions. The wealth of personal information present in speech signals and the availability of efficient techniques to identify that information pose a severe privacy risk for the users of speech interfaces. In particular, recent advances in voice cloning [307, 184, 215] and synthesis [282, 246] techniques that leverage "found speech" call for efficient speaker anonymization schemes.

Several open challenges pertaining to the privacy of speech data arise due to the emergence of large-scale data collection by voice-enabled apps and SPA devices. Privacy breaches by corporates like Samsung[4], Apple[5,6] and Google[7] have made headlines in the newspapers. The concern for malicious usage of such sensitive data has widely alarmed individual citizens, researchers and the legal commmunity. Recently, governments have also shown political will to support the efforts towards effective formulation of laws to achieve voice data protection by design and by default, along with the supporting technological advances that can secure the rights and interests of common citizens. With the above mentioned motivation, this thesis is a timely effort to propose speaker anonymization methods which aim to remove speaker identity from speech signals while keeping other linguistic attributes and speech quality intact. For widespread adoption of this technology, it is also important that the transformed speech remains usable for downstream tasks such as training an automatic speech recognition (ASR) model.

1.2 Scope and objectives

Although there have been a few efforts to protect speech signals against external attacks, the topic of privacy-preserving speech processing itself has attracted quite limited interest so far. The methods proposed in the last decade can be broadly classified into four categories: deletion, encryption, distributed learning and anonymization. Deletion refers to the blurring or obfuscation of sensitive segments of speech [45, 99] while retaining the acoustic scene, but has limited scope in terms of diverse speech applications. Encryption based methods [341, 31] aim to secure the transmission channel and perform operations in the encrypted domain, but incur a high computational cost and may require special hardware. Distributed learning methods such as federated learning [173] are machine learning techniques where training is performed by averaging gradients coming from several distributed nodes, thereby ensuring decentralization of training data to avoid central ownership of massive datasets, but may not protect the privacy of the speaker due to information leaking through gradients [94]. Finally, *anonymization* refers to the task of suppressing personally identifiable attributes of speech signal, leaving all other attributes intact. This thesis is a consolidated effort to propose effective methods and rigorous evaluation schemes for speaker anonymization. Hence we briefly review deletion, encryption and distributed learning based approaches here to present our arguments against using these approaches, thereby clearly defining the exact scope of our methods.

The earliest attempt at processing speech data in a private and secure manner mostly assumed a client-server model for speech applications, where the two parties communicate through mutually understood encrypted speech tokens and the transmission channel is secured by cryptographic methods, such as secure multiparty computation [266], secure two-party computation [22], hash functions and homomorphic encryption to represent and process speech [222], as well as nearest neighbour audio query search [239] or

[4] https://www.bbc.com/news/technology-31296188

[5] News article by The Guardian

[6] News article by The Bloomberg

[7] https://nos.nl/artikel/2292889-google-medewerkers-luisteren-nederlandse-gesprekken-mee.html

phonetic search [96] in the encrypted domain. Recently, with the introduction of cryptographic methods like homomorphic encryption and Paillier cryptosystem in neural networks [218, 227, 52], sensitive attributes in speech such as emotions have been identified in a secure manner [60]. Finally, Intel's[8] Software Guard Extensions (SGX) [48] provides a trusted execution environment in private regions of memory, called enclaves, for carrying out sensitive operations. VoiceGuard [31] presents a proof of concept by executing a speech recognition engine within the SGX enclave.

Although cryptographic methods have advanced manyfold, their strength is hinged upon the future breakability of the underlying encryption algorithm and the hardware resilience to adversaries. They require additional computational overhead and special hardware for successful implementation. Similar methods that protect privacy by reducing the speech signal to hash tokens destroy the paralinguistic and extra-linguistic characterstics of the signal, thereby losing all the utility for downstream tasks. Recent advances in federated learning [177] with the goal of decentralized ownership of user's data have enabled researchers to apply it to speech processing applications such as keyword spotting [173] and emotion recognition [165]. Although federated learning claims to protect users' privacy by not requiring them to share their data, it is not resistant to membership inference attacks [211]. Moreover, it has been shown that it is possible to reconstruct user's data given the knowledge of the received gradients [94].

This quick review of speech related privacy-preserving methods reveals that deletion, encryption or distributed learning based methods do not address the primary concern of this thesis, that is, to obtain an anonymous yet useful representation of speech with strong privacy guarantees. These methods do not focus specifically on the biometric speaker information present in the signal, instead they securely obfuscate the whole speech signal or devise a trusted data sharing mechanism. Hence, these methods do not align with our objectives, which are as follows: to recognise specific biometric identifiers present in the speech signal which makes it linkable to the speaker, to learn a *global transform* which could remove these identifiers effectively from the signal without affecting the linguistic content, and to evaluate the effectiveness of identity removal through strong attack measures and formal protection guarantees. Certainly, there have been earlier attempts to study speech transformation methods that claim to have removed speaker's identity up to a certain extent with varying loss of utility. We present an in-depth review of the research material on such anonymization techniques that closely align with our objectives in Sections 2.3 and 2.4.

In essence, we try to answer the following central question in this thesis:

> *While maintaining the usefulness of the signal, how to effectively remove the biometric identity of the speaker from any speech utterance?*

With the above stated central goal as the "holy grail" of privacy protection in speech, we reiterate that usability as well as privacy are the most important objectives of speaker anonymization. Here *usefulness* is a broad term that encompasses the ability to train models for downstream tasks such as ASR, human-level intelligibility for listening and transcription as well as the presence of the inherent varibilities of the natural speech signal. The intent to preserve these qualities of the speech signal while carrying out the process of anonymization emerges from the perspective of the potential users of this transformed speech data. Without the usability of the transformed speech corpus, the widespread adoption of speaker anonymization as the first step before speech data collection by digital assistant manufacturers and other service providers will not be possible. The hesitation to adopt anonymization techniques would directly lead to the non-compliance with the recent guidelines [78] put forth by the EDPB which cites the GDPR [77] and the e-Privacy Directive [76] to achieve privacy by design and by default while implementing and introducing virtual voice assistants in the market.

[8] https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html

We focus our efforts towards the development of speech transformation or representation learning based techniques to produce anonymous speech representations. This development aligns with our objective of identifying speaker-related information in the speech signal and leads us to explore different representations either in a client-server setting or independently of a fixed speech processing architecture, such as anonymous waveforms. We impose the constraint of being *global* over these tranformations so that speaker information is identified and removed across gender, accents, domains or recording conditions. We evaluate the privacy/utility trade-off of these representations in strong attack conditions, and prove their consistency through formal protection guarantees.

The combined goal of preserving usability along with privacy requires us to answer several fundamental questions. What constitutes the speaker's identity in a speech signal? How to identify and remove it without affecting the usefulness? To what extent does it make the speech signal linkable to the speaker? Can we disentangle speaker information from other attributes in the signal such as emotional states & traits, communicative acts, syntactic content, intonation, etc.? How to confirm the removal of identity among other attributes with high confidence?

Studies on speaker identification research have attributed speaker-related information to the human speech production mechanism [34]. Most of the speaker-related factors arise due to the physiology and shape of the vocal tract. Although the mechanism is well studied by linguists and the speech processing community, it is hard to formulate a global rule-based approach which modifies the speaker information alone and does not affect other attributes. In this thesis we explore machine learning based approaches which enable us to identify and remove speaker-dependent features in the speech signal either in the spectral domain or in learned features such as neural network representations.

Succinctly, the successful implementation of our central goal will translate not only into increased personal data protection but also increased trust by citizens and service providers. It will enable them to satisfy the requirements set by the law and eventually build society's trust in future private-by-design voice-based applications.

1.3 Summary of contributions

Our main focus is to obtain an anonymous speech representation which conceals the speaker's identity while retaining the linguistic content such that it can be used for further processing such as linguistic analysis, decoding the content (i.e. ASR) or training an ASR model. This representation of a speech utterance must be unlinkable to the person who spoke it, hence preventing an adversary to perform membership or linkage attacks. Clearly, there are many stakeholders in the process of anonymization, therefore we start by defining our attack model to concretize the goal of the anonymization process and demarcate the roles of the associated stakeholders in Chapter 3. We introduce the three actors affected by anonymization, namely: the speaker, the user and the attacker connected by an encompassing threat model focused on the speech data publication. In this chapter, we formally define the privacy and utility metrics that we use throughout the thesis and some preliminary experiments with a diverse set of voice transformation algorithms. We also compare three privacy metrics and investigate their usage based on the capacity to express vulnerabilities in anonymization. We shift the paradigm of speaker anonymization evaluation methods by formulating the premise of attacker's knowledge and gradually increase this knowledge to establish the idea of continuity from *Ignorant* to *Informed* attackers. All the subsequent representations are subjected to the rigorous evaluation regime proposed in this chapter and their performance is reported as per the established metrics.

We briefly consider a different threat model than the one mentioned in the previous paragraph. Unlike the previous model, it focuses on the privacy concerns surrounding the users of digital assistants. While recent studies have identified security vulnerabilities in these devices [171, 42], such studies tend to ignore

more important privacy risks that can have long-term impact. For instance, if the signal is intercepted by a malicious entity, a re-identitifcation attack can be launched against the user, potentially compromising the person's identity [241], intention [102, 115, 18, 275], gender [336, 160], emotional state [74, 306, 162], pathological condition [61, 302, 254], personality [252, 253] and cultural [255, 309] attributes to a great extent using state-of-the-art speech technologies. These algorithms require just a few tens of hours of training data to achieve reasonable accuracy, which is easier than ever to collect via digital assistants. The dissemination of voice signals in large data centers thereby poses severe privacy threats to the users in the long run. Hence, in Chapter 4 we propose a solution to prevent user's biometric identity information from leaving the digital assistants, thereby abating the growing privacy threats.

We present our experiments with adversarial learning to remove speaker information from the intermediate representation of the ASR network. Following the lead of previous approaches, we assume a client-server model, where private information is removed from the signal at the client side and the output is sent to server for decoding the text. We consider an end-to-end ASR model for this approach and the intermediate encoder representation is anonymized using speaker-adversarial learning. We evaluate the anonymous representations generated by this approach using an *Informed* attacker i.e. an attacker who possesses auxiliary knowledge about the anonymization mechanism. Such an attacker could easily diminish the strength of a weak anonymization process by learning the vulnerable discriminative patterns exhibited by the speakers after anonymization. The trend observed in the representations obtained after adversarial learning was that the anonymity does not genralize to unseen speakers. Moreover, this representation is tied to a fixed client-server architecture which restricts its usability to ASR by a fixed decoder server.

Given the rigid shortcomings of the client-server model and the intent to generalize the usability of anonymized representation to any arbitrary downstream task, we explore voice conversion techniques whose output is a speech waveform, i.e., an intelligible speech signal. Therefore, we again consider the threat model related to speech data publishing introduced in Chapter 3. More specifically, we experiment with x-vector based speaker anonymization where the speaker's identity is assumed to be perfectly disentangled from other factors of variation, such as the linguistic content and intonation, and concentrated only in the x-vector component. We extend the original idea and propose a baseline for the first VoicePrivacy challenge with flexible design choices to select the target identity for the source speaker. This approach is explained in Chapter 5. We conduct exptensive experiments with this approach and establish the superior privacy protection and utility achieved by it against *Semi-Informed* attackers, even in the worst-case scenario and in presence of thousands of speakers.

We also briefly analyse the usability of the anonymized speech data in Chapter 5. We first study the impact of re-training ASR models with anonymized data to ascertain whether they can perform similar to the baseline model when decoding the original speech samples. There was a clear gap between their performance due to lack of model generalization which is shown to be significantly reduced by augmenting the anonymized corpus with a small amount of original speech. The experiments exhibit that state-of-the-art acoustic models can be trained for ASR without requiring large scale un-anonymized (original) data. Such investigation repudiates the claim that original (untransformed) data is needed for training ASR systems.

Further experiments with the x-vector based anonymization framework reveal that the assumption of perfect disentanglement does not hold true, and in practice it is far from being perfect. The linguistic features and the prosodic pattern indeed retain some residual speaker information which makes the synthesized speech linkable to the original speaker even after anonymizaton. Moving forward, we drop the assumption of perfect disentanglement in x-vector based anonymization and undertake the task to measure the residual speaker information that might be present in the features extracted to represent the linguistic content and the prosodic pattern. We represent the linguistic content of an utterance using the bottelneck features (intermediate layers) extracted from the ASR network that can efficiently decode the textual content present

in the utterance. We assume that such bottleneck features capture the relevant phonetic information in the signal. The prosodic pattern is represented by the pitch (or fundamental frequency) contour which also demarcates the voiced-unvoiced regions of speech. We add Laplace noise to these features to make them differentially private and show that the anonymization can be improved by further removal of biometric identity from all the features. This approach is investigated in Chapter 6, where we first directly measure the individual contribution of pitch and bottleneck features towards speaker's identity and then exhibit superior privacy protection by formal noise addition.

To summarize, the contributions of this thesis are the following:

1. We define the attack model associated with privacy threats to speech interfaces. The three actors (the speaker, the user and the attacker) who are concerned with the anonymization task are mentioned and the anonymization techniques are evaluated from each of their perspective. These roles are analogous to real-world entities and the assumption of the knowledge they possess substantiates their capacity during the evaluation of anonymization schemes. Under the purview of this attack model, we propose a new regime of evaluation for speaker anonymization methods using the idea of *Informed* attacker. The attacker may possess auxiliary knowledge of the anonymization algorithm based on which he/she can design effective linkage functions to discover the true identity of the speaker. We establish the feasibility of such attacks by simulating several attackers with varying degree of knowledge about the anonymization scheme and show that previous studies have used an inferior model to evaluate their algorithms, namely the *Ignorant* attacker.

2. We conduct a comprehensive study of design choices for x-vector based speaker anonymization to select the target *pseudo-speaker* from an external pool of identities. This technique was also proposed as a strong baseline for the first VoicePrivacy challenge. We perform a complete analysis of the privacy/utility trade-off of each design choice in different attack scenarios as well as measure the sustainability of the best combination against re-identification when the attacker possesses the data of thousands of speakers.

3. We investigate the residual speaker-related information in the pitch contour and the bottleneck (BN) features and show that it can be removed through transformations or differentially-private noise addition. We also investigate the effect of pitch conversion on privacy and utility of the anonymized speech. We also propose an adversarial learning based transformation for BN features. Additionally, we put forth new neural network architectures for adding differentially-private noise to pitch as well as BN features and measure its impact of the privacy/utility trade-off.

4. We study the claim that the anonymized speech corpus is usable and propose techniques to train a viable ASR model which performs equally well for original and anonymized evaluation sets. Concretely, we explore a data augmentation based method to minimize the use of original data and generalize the performance of ASR models using mostly the anonymized speech corpus.

1.4 Publications

Publications as first author:

1. Brij Mohan Lal Srivastava, Aurélien Bellet, Marc Tommasi and Emmanuel Vincent. "Privacy-preserving adversarial representation learning in ASR: reality or illusion?" *In Proc. Interspeech*, pp. 3700–3704, 2019.

2. Brij Mohan Lal Srivastava, Nathalie Vauquier, Md Sahidullah, Aurélien Bellet, Marc Tommasi and Emmanuel Vincent. "Evaluating voice conversion-based privacy protection against informed attackers" *In Proc. 45th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2802–2806, 2020.

3. Brij Mohan Lal Srivastava, Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Junichi Yamagishi, Mohamed Maouche, Aurélien Bellet and Marc Tommasi. "Design choices for x-vector based speaker anonymization" *In Proc. Interspeech*, pp. 1713–1717, 2020.

4. Brij Mohan Lal Srivastava, Mohamed Maouche, Md Sahidullah, Emmanuel Vincent, Aurélien Bellet, Marc Tommasi, Natalia Tomashenko, Xin Wang and Junichi Yamagishi. "Privacy and utility of x-vector based speaker anonymization" *Submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

Other publications:

1. Mohamed Maouche, Brij Mohan Lal Srivastava, Nathalie Vauquier, Aurélien Bellet, Marc Tommasi and Emmanuel Vincent. "A comparative study of speech anonymization metrics" *In Proc. Interspeech*, pp. 1708–1712, 2020.

2. Natalia Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, J.-F Bonastre, Paul-Gauthier Noé and Massimiliano Todisco. "Introducing the VoicePrivacy initiative" *In Proc. Interspeech*, pp. 1693–1697, 2020.

3. Jean-Francois Bonastre, Hector Delgado, Nicholas Evans, Tomi Kinnunen, Kong Aik Lee, Xuechen Liu, Andreas Nautsch, Paul-Gauthier Noé, Jose Patino, Md Sahidullah, Brij Mohan Lal Srivastava, Massimiliano Todisco, Natalia Tomashenko, Emmanuel Vincent, Xin Wang and Junichi Yamagishi. "Benchmarking and challenges in security and privacy for voice biometrics" *In Proc. 1st ISCA Symposium on Security and Privacy in Speech Communication*, pp. 52-56, 2021.

4. Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Jose Patino, Brij Mohan Lal Srivastava, Paul-Gauthier Noé, Andreas Nautsch, Nicholas Evans, Junichi Yamagishi, Benjamin O'Brien, Anaïs Chanclu, Jean-François Bonastre, Massimiliano Todisco, Mohamed Maouche. "The VoicePrivacy 2020 Challenge: Results and findings" *Submitted to Computer, Speech and Language*, 2021.

5. Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Jose Patino, Brij Mohan Lal Srivastava, Paul-Gauthier Noé, Andreas Nautsch, Nicholas Evans, Junichi Yamagishi, Benjamin O'Brien, Anaïs Chanclu, Jean-François Bonastre, Massimiliano Todisco, Mohamed Maouche. "Supplementary material to the paper The VoicePrivacy 2020 Challenge: Results and findings" *Technical report*, 2021.

6. Ali Shahin Shamsabadi, Brij Mohan Lal Srivastava, Aurélien Bellet, Nathalie Vauquier, Emmanuel Vincent, Mohamed Maouche, Marc Tommasi, Nicolas Papernot. "Differentially private speaker anonymization" *Pending submission to USENIX*, 2022.

7. Mohamed Maouche, Brij Mohan Lal Srivastava, Nathalie Vauquier, Aurélien Bellet, Marc Tommasi, Emmanuel Vincent. "Enhancing speech privacy with slicing" `https://hal.inria.fr/hal-03369137/`, 2022.

1.5 Thesis structure

This thesis lies at the crossroads of speech processing, privacy and machine learning. In **Chapter 2**, we review the fundamentals and principles of these domains which are relevant for our work. First, we give a brief account of speech signal processing and artificial neural networks, then we describe the established tools of speech processing, such as speech recognition, speech synthesis and speaker identification. We recapitulate some key technologies that we employ while proposing our solutions, such as adversarial learning, speech transformation and voice conversion (VC). We review the existing methods of speaker anonymization which are close predecessors of our techniques. Thereafter, we formally define the anonymization task for which we propose the potential solutions.

In **Chapter 3**, we define the attack model and the actors involved in the process of anonymization. We firmly establish the concept of attacker's knowledge through preliminary experiments with voice conversion based techniques against *Informed* attackers. We present an in-depth comparative analysis of the metrics that are used for privacy evaluation.

Chapter 4 describes our effort to learn an anonymous representation of speech using adversarial learning which can processed locally and then transmitted to a server for decoding. We evaluate the privacy protection achieved by this representation using closet-set speaker identification as well as open-set speaker verification against an *Informed* attacker. **Chapter 5** introduces the VoicePrivacy initiative and the x-vector based speaker anonymization framework. We explore several design choices associated with it to choose a robust target *pseudo-speaker* x-vector. We conduct an extensive evaluation of the representations obtained using this technique in strong attack scenarios as well as against re-identification attacks in the presence of thousands of speakers. In this chapter, we briefly analyse the usability of the anonymized speech for training a state-of-the-art ASR system via data augmentation.

We further propose to measure and remove the residual speaker-related information in the inputs of the x-vector based speaker anonymization, i.e., the BN features and the pitch contour. These analyses are mentioned in **Chapter 6**. We add differentially-private noise in these features and measure the privacy/utility trade-off.

Finally, we summarize our contributions and reflect some of our perspectives towards the future directions opened up by the research done in the course of this thesis. **Chapter 7** concludes the thesis with these reflections.

Chapter 2

Background and Related Work

> If I have seen further it is by standing on the shoulders of Giants.
>
> *Isaac Newton*

There is abundant research material present in each of the domains relevant for this thesis. It uses previous knowledge from speech processing, privacy, and machine learning, and their associated sub-domains. In this chapter, we present selected background details of the sub-domains which are relevant to the techniques and terminology proposed in this thesis, and use the existing literature to elucidate the full picture of the problem statement and how the proposed solutions are perceived by the speech and privacy communities. We start by giving a brief account of the key historical advancements and current perspectives in speech processing and digital assistant technology that led to the crisis of privacy. We then describe the principles and tools of speech processing that will help the reader understand the basic terminology we use in the course of this thesis. Next, we present a brief review of the relevant literature which describes the previously proposed machine learning-based anonymization methods that are closely related to our work. Thereafter we formally define the task of privacy-preserving speech processing as presented in previous studies and how it was traditionally evaluated in terms of privacy and utility. Finally, we give a detailed explanation of the core techniques used in our proposed solutions, such as adversarial learning, voice transformation, and voice conversion.

2.1 A brief historical overview of speech processing and privacy

Speech processing came a long way since 1881 when the earliest device for recording speech was invented by Alexander Graham Bell. It used a rotating cylinder coated with wax over which up-and-down grooves could be cut by a stylus responding to the acoustic pressure generated by the sound wave. One can only imagine the tremendous challenges posed by this device to record, process, and store speech signals. Thankfully it has been replaced by microphones which capture the acoustic pressure from sound waves and record it as a relative change in voltage. There are several such historical advancements in speech technology that facilitated convenient and large-scale speech processing, eventually leading to the current privacy crisis. Particularly, Homer Dudley's work [66] inspired several generations of researchers to focus on making speech the mainstream medium for human-computer interaction, which propelled the large-scale storage of speech data and overall, the domain of speech signal processing forward.

Recall the "speaking machine" invented by von Kempelen introduced in Section 1.1 which could produce a few human-like sounds. In the mid-1800s, Sir Charles Wheatstone improved upon its design [321] using adjustable and configurable leather resonators capable of producing many more speech-like sounds. This model was adopted by Homer Dudley to design an electrical speech synthesizer [64] for Bell Labs. The synthesizer could be operated as a piano with hand controls to switch between voiced and unvoiced sounds, keys to control the characteristics of the signal and a foot pedal to control the pitch. It was called the VODER (Voice Operation Demonstrator) and was first demonstrated at the New York World's Fair in 1939. This event attracted the focus of researchers worldwide leading to several speech interest groups in the community. Dudley also pioneered the field of speech coding [270] which aims to represent speech signals for efficient storage and transmission by exploiting their inherent redundancies. He provided the analysis-synthesis [65, 67] method for speech coding.

The initial usage of speech technology was predominantly envisioned in a controlled setting, such as offices and research labs, where storage is limited, and through experience and training the people being recorded gradually became cautious to not divulge private information in the collected data. The recent advances that have led speech interfaces to enter our homes at the consumer level are quite new, and the privacy-related implications of this technology are still being explored. As of today, speech interfaces are present in personal mobile phones as well as digital assistants which have a widespread consumer base. Exposing an unaware user to such advanced technology will open the doors for potential adversaries to exploit the sensitive attributes present in the speech signal.

Several researchers have studied the security and privacy vulnerabilities of digital assistants [72, 166, 84], and their third-party applications [172]. The two most concerning privacy issues are the "always listening" feature and the cloud storage of the audio queries. The device remains in the inert state of buffering and re-recording until the wake word is spotted [135], it then records the audio and sends it to a cloud-based service for ASR and natural language understanding (NLU). All the audio files are usually stored in the user's account and can be accessed by logging into the account. This data may contain sensitive details about the user's life, such as bank details. A compromised account can lead to a user's private speech data being leaked to the public. Due to the rich nature of speech signal as we described earlier, not only the linguistic content but many other attributes of the speaker may become known to a malicious entity.

Extensive surveys of digital assistant users have been conducted to understand their mental models, beliefs, attitudes, and concerns towards their devices. Some studies [1, 166] show that users have an incorrect understanding of the working of digital assistants and the third-party services with which their sensitive data is shared. They are also unaware of the existing privacy controls in the digital assistant architecture. Malkin et al. [191] show that half of the users are not aware of the permanent retention policy of audio queries in the user's account. Users are not aware of existing privacy features and they express the need for automatic deletion of their recordings. Huang et al. [128] studied users' behaviour and privacy concerns when a digital assistant is shared among several housemates. Bispham et al. [27] present a taxonomy of attacks on speech interfaces which motivates future research on voice privacy to focus on exact vulnerabilities present in such devices. These surveys make some recommendations to users and manufacturers such as turning off the microphone when not in use, updating the firmware with the latest release, strict data deletion policies, and screening of sensitive content.

The above studies are indicative of the fact that users of speech interfaces are gradually becoming more aware of the underlying mechanisms and more concerned about their privacy being leaked through the interface. In this thesis, we aim to propose speaker anonymization techniques that will protect users' identity at the source, without requiring them to put in significant effort. These techniques can be built in directly into the device firmware by the manufacturers.

2.2 Principles and tools of speech processing

Now let us introduce some basic principles and tools of speech processing behind the proposed methods. We start with the basics of speech as a signal, and how it is processed to extract relevant features with physiological and phonetic considerations. We give a brief account of artificial neural networks due to their pervasive use as statistical models in speech processing tasks. Then, we describe the technology behind the three most popular speech applications that enable the design and evaluation of our proposed methods: automatic speech recognition, speech synthesis, and automatic speaker recognition.

2.2.1 Fundamentals of speech processing

In this section, we briefly discuss the mechanism of human speech production followed by its representation and processing as a discrete-time signal.

Vocal tract. The physiological apparatus that generates speech is called the *vocal tract* [112], which starts at the lungs and ends at the lips and the nostrils. The larynx (also called the voice box) separates the vocal tract into two anatomical regions: the lower part is called the sublaryngeal region and the upper part is called the supralaryngeal region. The sublaryngeal region of the vocal tract is composed of the diaphragm, the lungs, and the trachea (also called the windpipe). The air flows outward from the lungs and encounters a pair of flap-like structures in the larynx, called the *vocal folds*. When the vocal folds are held at an intermediate tension so that they are not too close or too far apart, the movement of the air induces ripples along their length. This causes them to vibrate, and the result is *voicing*. Voicing is the cause of periodic segments in the speech signal which are called *voiced* regions. On the contrary, when the vocal folds are held at sufficient distance from each other so that air flows freely through them, they do not vibrate, which results in voicelessness. This can be observed in the speech signal as aperiodic segments which look like random noise and are known as *unvoiced* regions.

The supralaryngeal region, which is composed of the oral cavity and the nasal cavity, plays a major role in determining the exact nature and quality of the sounds that are produced. The different parts of the supralaryngeal region that contribute towards the articulation of different vowels and consonants are referred to as articulators. The major articulators in the oral cavity are the lips, the teeth, the tongue, the alveolar ridge, the hard palate, and the velum. Among these, the tongue and the lower lip are the active articulators, whereas the others are passive and immobile. The complex interaction between active and passive articulators to completely stop the airflow, constrict it through a narrow channel, or allow it to pass through without restriction gives us the vast variety of speech sounds found in all of the world's languages.

Phonemes. The vocal tract is a continuous system capable of producing infinitely many sounds. These sounds are called *phones*. The exact physical mechanism of producing phones by the vocal tract, their transmission in acoustic space, and their auditory perception by the human ear are studied under the branch of linguistics called *phonetics* [112], which is independent of language. A given language can have only a small, finite number of sound units that can be used to compose words in that language and have some grammatical significance. These sounds must be perceivably distinct from each other for effortless communication and are called *phonemes*. The organization of phonemes, their combinations to produce words, and their semantic role in language are studied under the branch of linguistics called *phonology* [35]. Phonology categorizes the continuous signal produced by the vocal tract into discrete phoneme classes based on their acoustic, articulatory, and perceptual characteristics. Most languages feature two broad classes of phonemes, namely *vowels*, that are voiced sounds produced with no obstruction by the articulators, and *consonants*, that are produced by obstructing the airflow passing through the vocal tract. Although every language has a different

set of phonemes, the International Phonetic Alphabet (IPA) [267] describes the universal set of phonemes based on their articulatory characteristics.

Vowels are described based on the position of the tongue and the roundedness of the lips. The tongue is a highly active articulator and is subdivided into the front, central and back parts which can move somewhat independently of each other. It can also be placed at different heights to control the width of the constriction in the vocal tract. For example, /i/ as in "feed" is made by placing the front part of the tongue close to the hard palate, hence it is categorized as a close front vowel without rounding, while /o/ as in "foe" is made by raising the back of the tongue up to a certain height and rounding the lips, hence it is a close-mid back vowel with rounding. Consonants are categorized based on the presence or absence of voicing, the place of articulation that indicates the place of constriction in the vocal tract, and the manner of articulation which is the method of air release. For instance, /p/ as in "pan", is a voiceless consonant made by completely blocking the airflow using the lips, hence it is categorized as a voiceless bilabial plosive, whereas /z/ as in "zoo", is a voiced consonant produced by making a narrow constriction by placing the tip of the tongue close to the alveolar ridge, therefore it is a voiced alveolar fricative.

Such categorization of phonemes also helps us understand the linguistic behaviour of speakers when a sound is missing in their language [155, 163]. Generally, the non-native speakers retain voicing and manner, but replace the place of articulation, for example, the sound of consonant /ð/ as in the English word "the" is a voiced *dental* fricative that is not available in the French language, hence most native French speakers replace it with /z/ which is a voiced *alveolar* fricative [156]. Similarly, some dialects of Hindi do not have the phoneme /ʃ/ as in "sheep" which is a voiceless postalveolar fricative, hence they replace it with /s/ as in "sun" that is a voiceless alveolar fricative.

Speech in the time domain. Sound is a pressure wave traveling through the air as the medium of propagation. It can be recorded by measuring the variation in pressure at a single point in space over time. As mentioned before, a microphone is used to record the acoustic wave which measures the relative change in pressure as the electrical signal that is proportional to the pressure variation. Figure 2.1(a) shows the output of the microphone (for the word "privacy"), also called a waveform or a time-domain signal, pronounced by a male or a female speaker. The *duration* of both waveforms is shorter than one second. To represent a speech signal digitally, we must select the *bit depth* which is the finite precision needed to encode the amplitude values, and the *sampling rate* (denoted as F_s) which defines how many times per second the actual waveform is sampled to obtain discrete values of the amplitude. The duration, the bit depth, and the sampling rate decide the memory requirement to store the audio file. The audio file can be stored in a lossless uncompressed format (e.g., ".wav") or a lossy compressed format where the file size is reduced while maintaining good audibility (e.g., ".mp3").

A discrete-time speech signal can be represented as $\mathbf{s}$ and $\mathbf{s}[n]$ denotes a single sample of instantaneous amplitude value, where $n = 0, \ldots, N_{\mathbf{s}} - 1$. As described further, the speech signal is generally analyzed to determine its frequency components in a short duration.

Short-term analysis. A spoken sentence is also called an *utterance* which is a sequence of phonemes (note the phoneme annotations in Figure 2.1(a)). Depending on the recording conditions, whether the speaker is reading a given text or engaged in a spontaneous conversation, the utterance may or may not be grammatically correct. In any given utterance, except for global utterance-level information such as duration or speaker-related characteristics, other properties of a speech signal, like amplitude, voicing, etc. vary over time. We also know that different sounds are produced by different configurations of the articulators, so the system producing the signal itself is changing over time. Hence, in order to process the speech signal, it is divided into uniform regions called *time frames* that are individually analyzed. The speech signal $\mathbf{s}$ is

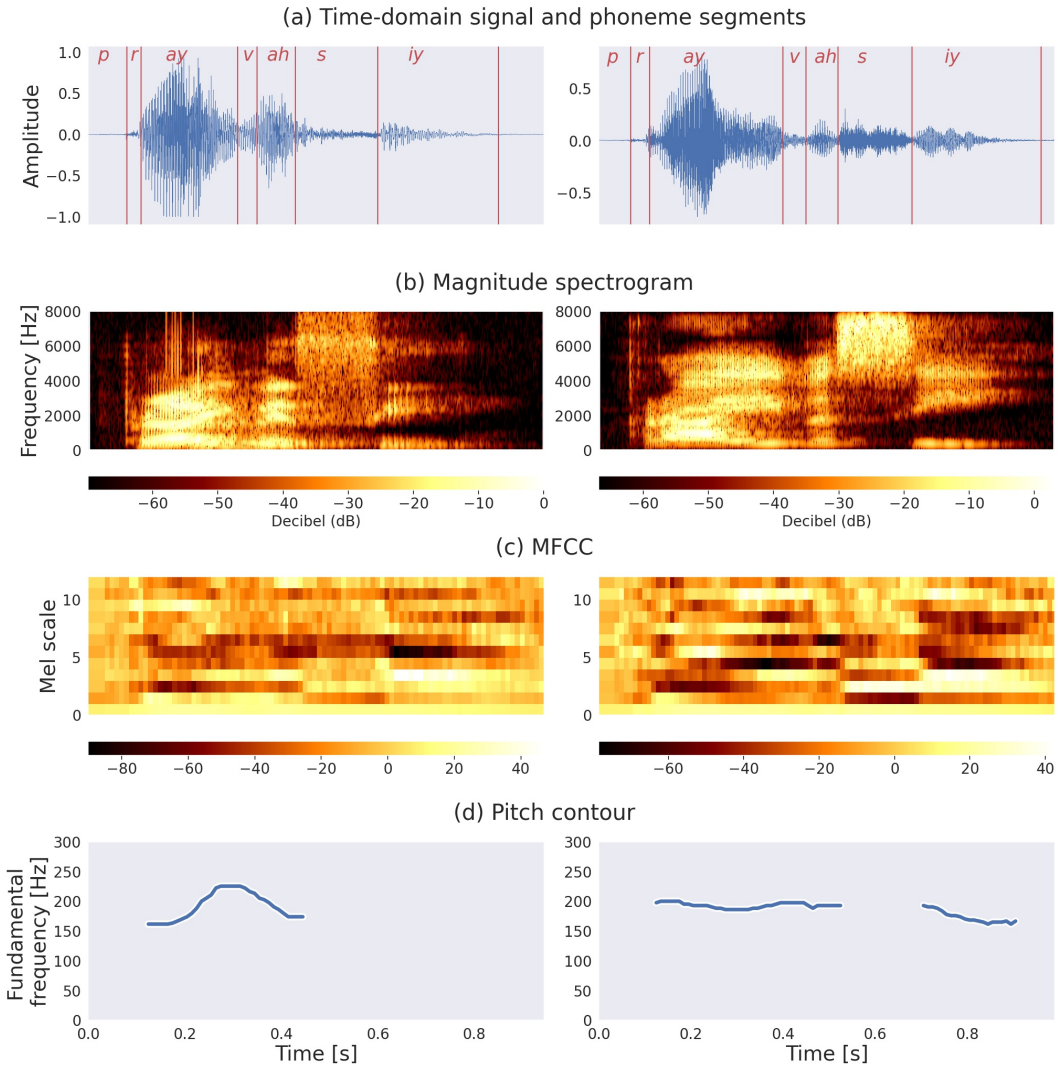


Fig. 2.1 Waveform, magnitude spectrogram, MFCC and pitch contour for the word "privacy", pronounced by a male (left) and a female speaker (right).

divided into T frames (subsequences of $\mathbf{s}$) each of length L samples with an overlap of $L/2$ samples, thereby obtaining $[\vec{s}_0, \ldots, \vec{s}_{T-1}]$, where for each $t \in [0, \ldots, T-1]$

$$\vec{s}_t = \left[\mathbf{s} \left[t \cdot \frac{L}{2} + l \right] \right]_{l=0}^{L-1}. \tag{2.1}$$

The time frames $\vec{s}_t$ are multiplied with a short window Ψ, of the same length as the time frame, i.e., L. The value of L is typically 25 to 30 ms. It is assumed that the system (vocal tract shape) remains stationary over the duration of the window, so that the signal's spectral properties are constant in this region. The window function [219] is not rectangular but tapering at the beginning and end, such as the Hamming window:

$$\Psi[l] = 0.54 - 0.46 \cos\left(\frac{l\pi}{L}\right), \tag{2.2}$$

to avoid introducing artifacts in the original signal. Due to the tapering window, we might lose some information present in the signal during analysis, hence a small overlap ($L/2$) is introduced between consecutive frames. The analysis of the speech signal after splitting it into a sequence of overlapping frames is called short-term analysis.

Speech in the frequency domain. By analysis, it is implied that we want to determine the frequency content of a particular frame of the speech signal. This can be done by faithfully reconstructing the signal as the weighted sum of simple sine waves called the basis functions. The basis functions are orthogonal to each other, i.e., no energy at the frequency of one sine wave is present in another. This property helps to get a unique solution for the coefficients of the weighted sum. Each basis function with unit amplitude and only a single fixed frequency is correlated with the signal to determine the exact magnitude of this frequency present in the signal. The frequency of the basis functions ranges from the lowest value, where a single cycle of the sine wave fits the entire analysis frame, to the highest possible frequency which is half of the sampling rate, also called the *Nyquist frequency* ($= F_s/2$). The process of computing the frequency content of the original time-domain signal is called the *Fourier transform*. The discrete Fourier transform of the t-th time frame $\vec{s}_t$ is computed to extract the Fourier coefficients $\mathcal{F}_t$ for the k-th frequency component:

$$\mathcal{F}_t[k] = \sum_{l=0}^{L-1} \vec{s}_t[l] \cdot \Psi[l] \cdot e^{-j\frac{2\pi}{L}kl}, \quad 0 \leq k \leq L-1. \tag{2.3}$$

The value of k corresponds to the frequency bin center $F(k) = kF_s/L$ in Hz (with zero at the start). The frequency bin centers are used to arrange the Fourier coefficients as meaningful frequency-domain representations, like a spectrum, and are also used to derive the features that warp the frequency axis, as described in Equation (2.5). Since $e^{j\omega} = \cos\omega + j\sin\omega$, $\mathcal{F}_t[k]$ is a complex number for each of the L frequency bands which encode the signal's magnitude and phase. This process can be sped up by the fast Fourier transform (FFT) [217] algorithm when L is a power of 2.

The vector of Fourier coefficients ($\mathcal{F}_t$) of a given frame is called a *spectrum* which has the size of $L \times 1$. For analysis purposes, it is common practice to record only the magnitude $|\mathcal{F}_t[k]|$ of the coefficients and discard their phase. The resulting vector is called the magnitude spectrum. Stacking the magnitude spectra of all frames results in a 2D representation of the whole utterance called the *magnitude spectrogram* (denoted as $|\mathcal{F}|$) which has the size of $L \times T$. The dynamic range is generally compressed by expressing the magnitude on a logarithmic scale called decibels (dB). Figure 2.1(b) depicts the magnitude spectrogram of the given speech signal.

Features. Frequency-domain representation has proven to be very powerful in order to inspect the properties of speech sounds. For example, the smooth curve that follows the peaks of the spectrum for any given analysis frame is called the *spectral envelope*, and it is governed by the shape of the vocal tract. The dominant peaks corresponding to the resonant frequencies in the spectral envelope are called *formants*. Each phoneme is characterized by specific spectral properties which can be used as a template to recognize it [116]. Using the spectrum or the spectral envelope directly for speech recognition may however not be optimal due to the inherent covariance between frequency bands, and the range of magnitude which does not linearly correspond to loudness.

To alleviate these shortcomings, researchers have proposed several transformations of the spectrum such that the resulting features correspond to a compressed representation which is motivated by the perceptual mechanism of the human ear [141]. It is well known that humans can perceive sound within a defined frequency range of 20 Hz to 20 kHz. The human auditory system is more discriminative between tones at

lower frequencies and increasingly less discriminative at higher frequencies [179]. The sensitivity to higher frequencies also reduces as we age. Hence speech signals are generally processed at the sampling rate of 16 kHz or lower, limiting the information to 8 kHz based on the Nyquist frequency. After the FFT, the magnitude spectrum $|\mathcal{F}_t|$ is obtained for each frame, which is then warped according to a non-linear perceptual scale called the *Mel scale*. The linear frequency F (in Hz) can be converted to Mel scale using the following formula:

$$\mathrm{Mel}(F) = 1127 \cdot \ln(1 + F/700). \tag{2.4}$$

The Mel scale aims to mimic the sensitivity of the human auditory system by warping the frequency scale using closely-spaced narrow bandpass filters at lower frequencies, and increasingly wider and sparsely-spaced filters at higher frequencies.[1] The Mel scale filters capture the general shape of the spectral envelope needed for speech recognition and smoothes out the harmonics,[2] thereby losing the fundamental frequency information. The warped magnitude spectrum is segmented into frequency bands according to a Mel filter bank which consists of a fixed number of overlapping triangular filters, typically 40 to 80, defined by their center frequencies $F_c(m)$ on the linear scale. The Mel filter bank is parameterized by the number of filters N_M, the minimum frequency $F_{\min}$, and the maximum frequency $F_{\max}$. The center frequencies of the Mel filters are the integer multiples of the fixed frequency resolution δ_{Mel} in the Mel scale which is computed using $\delta_{\mathrm{Mel}} = (\mathrm{Mel}(F_{\max}) - \mathrm{Mel}(F_{\min}))/(N_M + 1)$. Hence, the center frequencies are given by $\mathrm{Mel}(F_c(m)) = m \cdot \delta_{\mathrm{Mel}}$ for $m = 1, \ldots, N_M$. The center frequencies of the triangular filters are converted to the linear scale using the inverse mapping: $F_c(m) = 700 \cdot (e^{\mathrm{Mel}(F_c(m))/1127} - 1)$. The Mel filter bank $M(m, k)$ is given by [158]:

$$M(m,k) = \begin{cases} 0 & \text{for } F(k) < F_c(m-1), \\ \frac{F(k) - F_c(m-1)}{F_c(m) - F_c(m-1)} & \text{for } F_c(m-1) \le F(k) < F_c(m), \\ \frac{F(k) - F_c(m+1)}{F_c(m) - F_c(m+1)} & \text{for } F_c(m) \le F(k) < F_c(m+1), \\ 0 & \text{for } F(k) \ge F_c(m+1). \end{cases} \tag{2.5}$$

The Mel filter bank $M(m, k)$ is a matrix of size $N_M \times L$ which, when multiplied by the power spectrum (i.e., squared magnitude spectrum), yields a set of coefficients called the *Mel-filterbank coefficients*. To further enhance their usability, a logarithm is applied to compress the dynamic range such that it is more directly related to the perceptual loudness. The resulting logmel coefficients are sometimes directly used for speech recognition [100] and synthesis [143]:

$$\mathrm{logmel}_t(m,k) = \ln \left\{ \sum_{k=0}^{L-1} M(m,k) \cdot |\mathcal{F}_t[k]|^2 \right\}. \tag{2.6}$$

Finally, the discrete cosine transform (DCT) can be applied to these coefficients to approximately de-correlate them from each other and obtain *Mel-frequency cepstral coefficients* (MFCCs) as shown in Figure 2.1(c). MFCCs are widely used in speech applications. Note that logmel or MFCC are real-valued vectors obtained per frame: they are often concatenated over time to get the whole Mel spectrogram or MFCC sequence for an utterance.

[1] The Mel scale is the most popular perceptual scale, but there are other such scales like the Bark scale.

[2] A periodic signal with frequency F is only composed of the frequencies that are integer multiples of F, i.e., F, $2F$, $3F$, etc. These frequencies are called harmonics.

Pitch. An important property of speech is the presence of *pitch* in the voiced regions. Strictly speaking, pitch is the perceptual property that relates to the rising-falling tonal pattern, or the intonation of speech. It highly correlates with a physical property of the speech signal, called the fundamental frequency (denoted as $\mathbf{p}$), which is the rate of vibration of the vocal folds. The range of pitch is determined by the physiological factors of the vocal folds, such as their mass and length, hence it depends on the speaker and is typically lower for male than female [25]. The pitch sequence governs the *intonation* of the spoken utterance, and it significantly contributes towards the message that is being conveyed to the listener; for instance, a rising pitch at the end of the sentence may convey to the listener that a question is being asked. It is a key component of prosody (together with stress and rhythm) which determines the utterance expressiveness, and is crucial for speech synthesis. Prosody is a useful tool of communication in language as it indicates the prominence of different linguistic units that compose the utterance, and hence contribute towards the naturalness of speech. It is important to note that pitch is the rate of vibration of the vocal folds hence it is only defined for voiced phonemes such as /a/, /b/, /z/, etc. It is pointless to compute pitch for silence, noise or unvoiced regions of an utterance since there is no vibration of the vocal folds, hence by convention the pitch value in these regions is set to zero.

Pitch can be estimated from the speech signal, without having physical access to the vocal folds, using pitch estimation algorithms [264]. Pitch estimation is a difficult task due to erroneous observation of harmonics causing pitch doubling/halving [332]. It is also difficult to estimate the pitch when the quality of speech is distorted due to noise or channel effects. A fairly robust and widely used algorithm for pitch tracking is called Yet Another Algorithm for Pitch Tracking (YAAPT) [148]. It is a hybrid pitch tracking method as it considers both the time and the frequency domain to estimate the value of $\mathbf{p}$. It comprises a nonlinear preprocessing step on the squared speech signal, followed by $\mathbf{p}$ estimation using Spectral Harmonics Correlation from the spectrogram of the nonlinearly processed signal. A crucial components of YAAPT is the normalized cross-correlation function (NCCF) which is used to extract prominent peaks corresponding to $\mathbf{p}$ candidates in the time domain. The NCCF for a given time frame and lag-index q is defined as [283]:

$$\mathrm{NCCF}_t(q) = \frac{1}{\sqrt{\xi_0 \xi_q}} \sum_{l=0}^{L-Q} \vec{s}_t[l]\vec{s}_t[l+q]. \tag{2.7}$$

The NCCF_t is computed for $0 \leq q < Q$, where the value of maximum lag Q is generally lesser than the frame length L, and the frame energy is given by $\xi_q = \sum_{l=q}^{q+L-Q} (\vec{s}_t[l])^2$. The final pitch contour is obtained using dynamic programming and a normalized low frequency energy ratio function is applied to make voiced/unvoiced decision. Figure 2.1(d) shows the estimated pitch for the word "privacy" produced by a male or a female speaker.

2.2.2 Artificial neural networks

At this point, we digress a little bit to explain the core concepts of artificial neural networks (ANNs) and deep learning, which form the key components of modern speech processing tasks, as we will see later in this chapter. ANNs are inspired by the working and structure of the biological neural network present in the brain. The idea of ANNs emerged from the *connectionist* [93] school of cognitive science which hopes to simulate human intelligence through a large network of connections between the neurons. Neurons are the smallest unit in ANNs and are represented as nodes in this large computational graph. Actions are triggered when a specific combination of neurons are fired together. ANNs derive their power from the general framework proposed in the seminal work of parallel and distributed processing [247], which describes the parallel nature of neural information processing and the distributed nature of neural representation in ANNs, which are similar to how the brain processes and stores information. Although we are still not

aware of the complete working of the human brain, storage of memories, production of thoughts, etc., we know that there are special regions for processing different types of sensory inputs, such as the visual and auditory cortex. Thus, ANNs are typically used to learn specific tasks, such as object detection or phoneme recognition, which they can accomplish quickly and sometimes more precisely than human beings, rather than designing a general-purpose intelligent machine, like the brain.

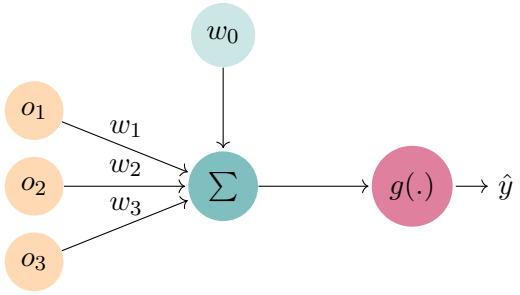


Fig. 2.2 Perceptron model of a neuron with $A = 3$.

Structure As mentioned before, a neuron is the fundamental unit of an ANN, sometimes also called a *perceptron* model [243]. It is nothing but the weighted sum of its inputs, which is transformed by a non-linear *activation function*, denoted by $g(.)$ as shown in Figure 2.2. The activation function decides whether the neuron fires or not based on the cumulative influence of the input features and the activation threshold, thereby filtering the information that is passed on to the output. The activation function should be non-linear so that the neuron can learn the complicated non-linear relationship between the input and the output. It should also be differentiable so that the gradients of the error can be computed and *backpropagated* to optimize the model parameters, θ. The output activation of a perceptron can be simply written as:

$$\begin{aligned} \hat{y} &= g\left(w_0 + \sum_{i=1}^{A} w_i o_i\right) \\ &= g(\theta^T \mathbf{o}). \end{aligned} \tag{2.8}$$

Here, we define $\mathbf{o} = [1, o_1, \ldots, o_A]$ as a single sample (observation) from the data set containing A features and an additional 1, and the parameters $\theta = \{w_0, w_1, \ldots, w_A\}$ include one synaptic weight w_i for each feature and a bias w_0 to ensure that the decision boundary isn't fixed at the origin. In real-world applications, we encounter complex multi-class problems such as speaker identification, phoneme recognition, etc. that require better expressivity and the ability to learn complex non-linear mappings/representations. Therefore, in practice we use neural networks with a more complicated architecture than a perceptron and several interconnections that exist between millions of neurons represented by the weights and biases. The input is propagated forward sequentially through the consecutive layers, where each *layer* contains multiple neurons. This is referred to as *forward propagation*, which is used to compute the output of the neural network.

A *fully-connected* network involves directed connections from each neuron in the current layer to every neuron in the subsequent layer as shown in Figure 2.3. It is sometimes also referred to as a *multilayer perceptron*. The first layer which receives the features directly is called the *input layer*, and the last layer is called the *output layer*. The remaining layers in between are called *hidden layers*. The neural network shown in Figure 2.3 includes two hidden layers, where $h_j^{(k)}$ represents the activation value for the j-th neuron in the k-th layer. Although it has been proved that a neural network with a single hidden layer and enough

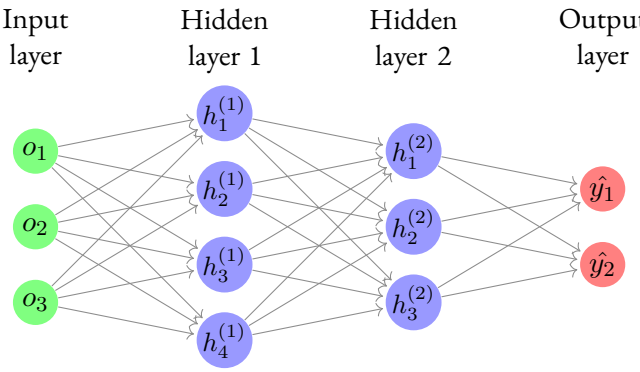


Fig. 2.3 Fully-connected feed-forward neural network (multilayer perceptron).

neurons can approximate any computable function [51], it is more costly to add neurons in a single hidden layer than to add more hidden layers. Moreover, it has been repeatedly shown that the sequential hidden layers learn representations of data with multiple levels of abstraction [167], which gave rise to the field of *deep learning*.

Training Let there be input samples $\{\mathbf{o}_i\}_{i=1}^N$ in a given data set and the corresponding desired outputs $\{y_i\}_{i=1}^N$ where $\mathbf{o}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Then, an ANN, like most machine learning algorithms, can be simply considered as a non-linear function $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$ with some set of parameters θ, which maps an input sample $\mathbf{o}_i$ to an output y_i. Although not limited to, among other tasks ANNs are generally used to solve two types of problems in machine learning: regression and classification. When y_i takes continuous values, such as commodity prices, the size of a tumor, spectral amplitudes, etc., the task is a *regression* problem. On the contrary, when y_i is a class within a discrete set, such as speaker identities, phonemes, tumor presence or absence, etc., it is a *classification* problem.

ANNs learn the mapping from $\mathcal{X}$ to $\mathcal{Y}$ up to a reasonable bound of accuracy by adjusting their parameters θ based on the total amount of error between the ground truth y_i and the estimate $\hat{y}_i$. The function that measures this error is called by different names: the *cost*, the *loss* or the *objective* function, denoted by $\mathcal{L}(\theta)$. Indeed, the value of θ determines the current value of $\mathcal{L}$, and the goal of training algorithms is to minimize the value of $\mathcal{L}$ until convergence. The loss function used for real-world problems is usually a non-convex function of θ [343]. If $\mathcal{L}$ is differentiable, then we can make a step in the steepest direction by simply finding the gradient of $\mathcal{L}$ with respect to each element in θ over the whole data set, subtract the gradient $\nabla\mathcal{L}$ from corresponding element in θ iteratively to nudge it in the direction where $\mathcal{L}$ is smaller. The gradients are generally scaled by a small value η, called the *learning rate*, which decides the step size to avoid missing the minimum when it is too close. The process of analytically computing the partial derivative of the error with respect to each parameter, and updating those parameters to minimize the loss function is the workhorse of machine learning and is referred to as *backpropagation* using *gradient descent* [245].

A commonly used loss function is the mean squared error (MSE) which measures the average squared distance between the desired and the actual output of the neural network:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2. \tag{2.9}$$

A step in gradient descent to update each parameter θ_j is given as follows:

$$\theta_j \leftarrow \theta_j - \eta \frac{\partial \mathcal{L}}{\partial \theta_j}. \tag{2.10}$$

In practice, the huge size of the training data set makes it intractable to compute the gradient over the whole data set at once as shown in Equation (2.9). Alternatively, one may compute gradients over individual training examples and update the parameters each time, this is called *stochastic gradient descent* (SGD). Each step of SGD can be computed much faster than regular gradient descent, but the variance of updates is much higher, leading to fluctuations in the loss function. As a trade-off between the two methods, the gradients can be computed over disjoint subsets of training data called *mini-batches*. Computing the gradient over a mini-batch is computationally efficient and leads to more stable convergence. This modified version of the algorithm is called the *mini-batch gradient descent*, but it is interchangeably referred to as SGD in the literature so we will call it SGD from here on. When SGD has seen the whole training set, i.e., it has computed and backpropagated the gradients over all the mini-batches, this is referred to as the completion of an *epoch*. For better results, systems are trained for several epochs until the loss does not change significantly any further.

Activation function There are several choices for the activation function, such as the sigmoid function, hyperbolic tangent (tanh), rectified linear unit (ReLU), softmax, etc. The sigmoid function was traditionally used because it squashes inputs to the $[0, 1]$ range, but its derivative is upper bounded by 0.25 which implies that the magnitude of the gradient values reduces by at least 75% at each layer. This leads to the *vanishing gradient* problem [284] in deep networks, which also arises with the tanh activation function. Hence, in recent years ReLU [334] has become the preferred choice of activation:

$$g(x) = \max(x, 0). \tag{2.11}$$

The derivative of the ReLU function is as follows:

$$g'(x) = \begin{cases} 0 & \text{for } x < 0, \\ 1 & \text{for } x \geq 0. \end{cases} \tag{2.12}$$

The softmax function is another popular activation function that ensures that the outputs are positive and that they sum up to 1. It is commonly used as the activation for the output layer in classification problems. The outputs can then be interpreted as a probability distribution over the categorical classes.

Relevant deep neural network models Deep learning has enabled researchers to explore several complex network architectures which may be suitable for specific tasks. Here we briefly discuss some of the models that are relevant for processing speech data. As described in Section 2.2.1, speech data is processed by decomposing an utterance into fixed-length overlapping segments called frames. In applications such as speech recognition or speaker identification, MFCC or logmel features are computed for each frame, and the whole sequence is fed as input to a neural network that accounts for the temporal dynamics of speech.

The simplest of all deep neural network (DNN) architectures is the *feed-forward* network as shown in Figure 2.3. Simple feed-forward architectures are great function approximators for data that can be represented by independent factors, such as predicting loan application outcomes using a person's financial attributes, but they fail to efficiently capture the local spatial and temporal relationships that exists in image

and speech data. Hence other architectures, such as convolutional neural networks (CNNs) [168], have been proposed to model these relationships effectively.

CNNs are a special type of feed-forward network, which is best suited for image data since it is inspired by the working of the visual cortex. The fundamental neuron in a CNN acts like a *kernel* having a pre-specified 2D receptive field, that moves over the input image and performs a convolution operation with the area covered by it. Several kernels in a layer convolve with the same input image to learn different spatial properties of the image. They are followed by a non-linear activation and a pooling operation to transform the kernel output into a *feature map*. At each successive layer, the kernels learn to discriminate between hierarchical features such as edges, geometrical shapes, objects and eventually lead to a fully-connected layer that predicts the output classes.

Another feed-forward architecture was proposed to model the temporal dependencies present in speech data, called the time delay neural network (TDNN) [311]. It can also be seen as a CNN with 1D kernels, where each layer operates at a different temporal resolution as shown in Figure 2.4. The bottom layers of a TDNN learn an affine transform for a narrow context window at each time step, and the context becomes wider in upper layers. Due to shared kernel weights across time steps, TDNNs are capable of learning translation invariant feature transforms. It has been observed that TDNNs can be made computationally efficient by sub-sampling the activations that are passed on to the next layer due to a large overlap between neighbouring contexts [224]. This process of sampling non-contiguous frames for building the context is called *dilation*.

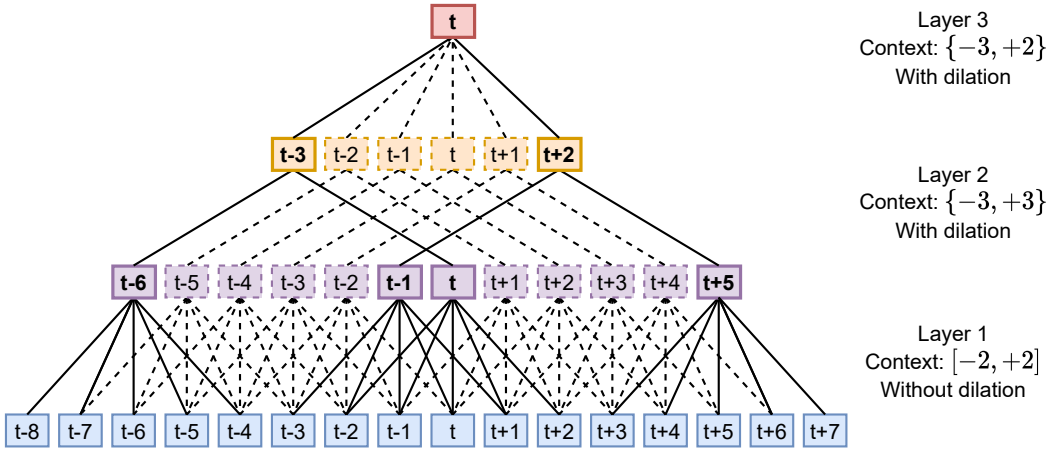


Fig. 2.4 Time delay neural network architecture with dilation in layers 2 and 3. Dotted lines indicates the connections and the nodes which are not included in the computation due to dilation applied to the layers.

Factorized TDNN (TDNN-F) models as depicted in Figure 2.5 have been proposed by Povey et al. [229] to reduce the number of parameters and the computational cost. Each unit of a TDNN hidden layer acts as a 1D kernel which produces a feature map by processing several time frames together depending on the temporal resolution of the layer. Let $\mathbf{W}$ be the weight matrix between the hidden layer and the feature map, then TDNN-F factorizes $\mathbf{W} = \mathbf{PQ}$ into two factors using the singular value decomposition, and imposes a constraint such that $\mathbf{P}$ is semi-orthogonal, i.e., $\mathbf{PP}^\top = \mathbf{I}$ or $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$. The interior dimension between $\mathbf{P}$ and $\mathbf{Q}$ is much smaller than the number of units in the hidden layer or the feature map and it is referred to as the linear bottleneck dimension. It is assumed that even with a reduced number of parameters, no model strength is lost if one of the factors is constrained to be semi-orthogonal. This contraint is imposed every four training iterations by updating matrix $\mathbf{P}$ such that it is closer to being semi-orthogonal by using the

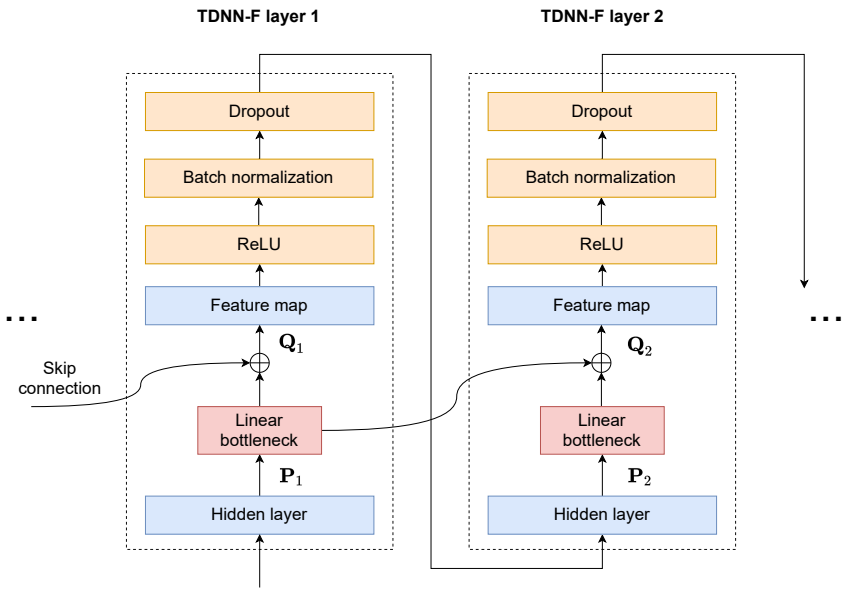


Fig. 2.5 Factorized TDNN (TDNN-F) architecture showing the linear bottleneck inserted between the hidden layer and the feature map. Matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ are constrained to be semi-orthogonal and $\oplus$ represents the concatenation of the linear bottleneck and the feature map layer linked by the skip connection.

following rule:

$$\mathbf{P} \leftarrow \mathbf{P} - \frac{1}{2\eta^2}(\mathbf{P}\mathbf{P}^\top - \eta^2\mathbf{I})\mathbf{P}. \tag{2.13}$$

Here η modifies the constraint such that a scaled version of $\mathbf{P}$ is expected, and it is similar to the learning rate hyperparameter of a neural network because it controls how fast the layer parameters are changing in a consistent manner. It is claimed that a network composed of TDNN-F layers does not require pretraining, but the training may be unstable if $\mathbf{P}$ is too far from being semi-orthogonal. Hence, it is initialized using the Glorot mechanism [97] and the learning rate is carefully chosen as $\eta = \sqrt{\mathrm{tr}(\mathbf{K}\mathbf{K}^\top)/\mathrm{tr}(\mathbf{K})}$, where $\mathbf{K} \equiv \mathbf{P}\mathbf{P}^\top$ and $\mathrm{tr}(\cdot)$ computes the trace of a matrix. Another feature that helps stabilize the training of TDNN-F networks is *skip connections* which append[3] the linear bottleneck of previous layers to the feature map of the current layer as shown in Figure 2.5. Such a network is much faster to train using parallel computing (GPUs) than other neural networks which model temporal dependencies, such as recurrent neural networks (RNN), due to their feed-forward architecture.

TDNNs model temporal dependencies by merging contextual information with the input at the current time step and estimating the output through a feed-forward mechanism. In contrast, RNNs do not follow a feed-forward mechanism, but incorporate some kind of memory or *hidden state* of a sequence that remembers information in previous time steps and is used for subsequent computations. The hidden state for the current time step is obtained by combining the hidden state in the previous time step and the current input. The parameters of current and previous hidden states are optimized based on the feedback from the current output using a modified version of backpropagation, called "backpropagation through time" [320]. One limitation of vanilla RNNs is that they can be very inefficient at learning relevant information in the sequence due to gradients vanishing across time, hence several RNN variants have been proposed to retain

[3] Generally, a skip connection *adds* the input of the current layer (or a previous layer) to the output of the current layer, but [229] refers to concatenation as the skip connection.

the feedback signal, based on long short-term memory (LSTM) [118], bidirectional LSTM [101] (BLSTM) or gated recurrent unit (GRU) [43] layers. Another limitation is the sequential nature of computation at each time step which cannot leverage the enormous parallelism offered by advanced computing infrastructure like GPUs. Due to these limitations, RNN architectures are increasingly being replaced by convolutional or Transformer [305] based architectures for sequential data like speech and natural language. We will describe some of these architectures in later sections when we apply them to our applications.

Although applying DNNs to speech data has undeniably benefitted the community by improving the performance of the systems, it has significantly raised the demand for large-scale data speech collection. Previous studies have shown that without requiring new data, neural networks can avoid overfitting if the speech signals in the training set are artificially augmented with reverberation or noise [204, 157]. This way, the existing data set can be multiplied several folds with diverse settings contributing to the enrichment and robustness of the model. The experiments performed in this thesis rely on these techniques to achieve state-of-the-art results.

2.2.3 Automatic speech recognition

Automatic speech recognition (ASR) aims to convert an utterance into its textual content, also called transcription. The output of ASR is used by natural language understanding systems that take speech as input, and is widely deployed in commercial applications ranging from cloud servers to mobile devices. We mentioned before that speech utterances are of varying duration and the system producing them also varies through time, hence they are processed as a sequence of T overlapping time frames of fixed duration. The input to ASR is a sequence represented as a matrix, $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]^\top \in \mathbb{R}^{T \times A}$ of length T time frames, where $\mathbf{o}_t \in \mathbb{R}^A$ are feature vectors derived from the speech signal, e.g., MFCCs or logmel spectra, and the output is the estimated word sequence $\hat{W}$. This problem can be formulated as [330]:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|\mathbf{O}). \tag{2.14}$$

Researchers in the domain of ASR have tried to solve the problem defined in Equation (2.14) through two main approaches. The first one is called the *conventional HMM-based* pipeline and the more recent one is the *end-to-end ASR* approach. We use both of them in different parts of this thesis, therefore we give a brief overview of both of them below.

Conventional approach In this approach, it was noted that it is infeasible to directly model the conditional distribution of the most probable word sequence given the acoustic features. To simplify this problem, $P(W|\mathbf{O})$ can be decomposed into a simpler probabilistic model by defining a generative process, and then the true word sequence is inferred from it. The generative model is depicted in Figure 2.6 and defined as follows: we know that an utterance is a sequence of spoken words that are distributed according to the language model. Spoken words are in turn made up of a sequence of fundamental sounds called phonemes (ρ), but the same phoneme can manifest itself differently in the signal due to natural variation of the vocal tract and the context surrounding it, also known as the coarticulation effect. The different manifestations of a phoneme in the presence of varying contexts can be represented by triphones (i.e., tied context-dependent phonemes) where each triphone is modeled by its own hidden Markov model (HMM) and the speech features follow a Gaussian probability density function within each state. This is called the GMM-HMM approach. Alternatively, deep neural networks, instead of GMMs, can be used to model the density of HMM states [117], which is referred to as the hybrid DNN-HMM approach that is used in this thesis and described further. To handle data scarcity issues, the triphone HMM states are clustered using a decision

tree and the same emission probability is shared by all the states in a given cluster S, which is also called a "tied state". Hence, the ASR problem is reformulated as [192]:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(\mathbf{O}|W)P(W) \tag{2.15}$$

$$\approx \underset{W}{\operatorname{argmax}} \sum_{S,\rho} P(\mathbf{O}|S)P(S|\rho)P(\rho|W)P(W). \tag{2.16}$$

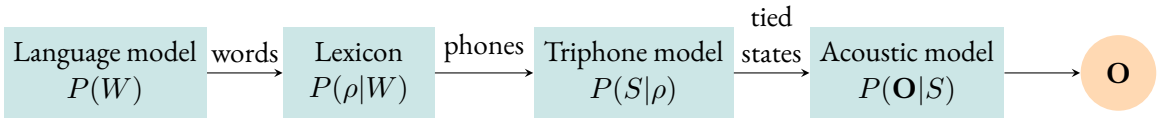


Fig. 2.6 Generative model for ASR.

Here $P(W)$ is the so-called language model that represents the prior distribution of word sequences, $P(\rho|W)$ is the lexicon which maps all the words in the vocabulary to their corresponding phoneme sequences, $P(S|\rho)$ maps a phoneme sequence to the corresponding tied state sequence $S = [S_1, \ldots, S_T]$, and $P(\mathbf{O}|S) \propto \prod_{t=1}^{T} P(S_t|\mathbf{O})/P(S_t)$ where the tied state posterior probabilities $P(S_t|\mathbf{O})$ are given by the so-called DNN acoustic model and $P(S_t)$ is the prior probability of each tied state. These models are learned independently and composed together as a graph using finite state transducers (FST). In some use-cases, a sequence of phonetic features called bottleneck (BN) features, denoted as $\mathbf{B}$, can be extracted from an intermediate layer of the ASR acoustic model [331] and used, possibly in combination with other features, for other tasks. The above mentioned generative model is also used to "synthesize" speech utterances as explained in the next section.

The DNN acoustic model $P(S_t|\mathbf{O})$ is trained on acoustic features $\{\mathbf{O}_i\}_{i=1}^{N}$ extracted from the utterances $\{\mathbf{s}_i\}_{i=1}^{N}$ in some annotated data set $\mathcal{D}$ and the corresponding transcriptions $\{W_i\}_{i=1}^{N}$. The cost function $\mathcal{L}_{\text{ASR}}$ which can be carefully crafted to predict the accurate triphone sequence, is minimized to optimize the parameters of the acoustic model. For example, one popular [123, 108, 95] cost function is the following: $\mathcal{L}_{\text{ASR}} = \mathcal{L}_{\text{MMI}} + 0.1 \cdot \mathcal{L}_{\text{CE}}$, which is composed of two terms. The dominant term, $\mathcal{L}_{\text{MMI}}$, is the lattice-free maximum mutual information (LF-MMI) [231] cost which aims to maximize the posterior probability of the ground truth word sequence W_i:

$$\mathcal{L}_{\text{MMI}} = -\sum_{i=1}^{N} \log \frac{P(\mathbf{O}_i|W_i)P(W_i)}{\sum_{W'} P(\mathbf{O}_i|W')P(W')}. \tag{2.17}$$

The numerator is the joint likelihood of the acoustic features $\mathbf{O}_i$ and the ground truth word sequence W_i, while the denominator is the likelihood of the acoustic features marginalized over all possible word sequences. The numerator is computed by summing over all tied state sequences corresponding to W_i: $P(\mathbf{O}_i|W_i) = \sum_{S_i,\rho_i} P(\mathbf{O}_i|S_i)P(S_i|\rho_i)P(\rho_i|W_i)$ where $P(\mathbf{O}_i|S_i) \propto \prod_{t=1}^{T_i} P(S_{i,t}|\mathbf{O}_i)/P(S_{i,t})$, and $P(S_{i,t})$, $P(S_i|\rho_i)$, $P(\rho_i|W_i)$ and $P(W_i)$ are fixed. The numerator is computed in a similar way, except that the (intractable) sum over all possible word sequences with a word-level language model is approximated by a (tractable) sum over all possible phoneme sequences with a phoneme-level language model. The second term $\mathcal{L}_{\text{CE}}$ of the cost function is the frame-level cross-entropy loss between the true and estimated tied states, which acts as a regularizer [231]:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N}\sum_{t=1}^{T} \log P(S_{i,t}|\mathbf{O}_i). \tag{2.18}$$

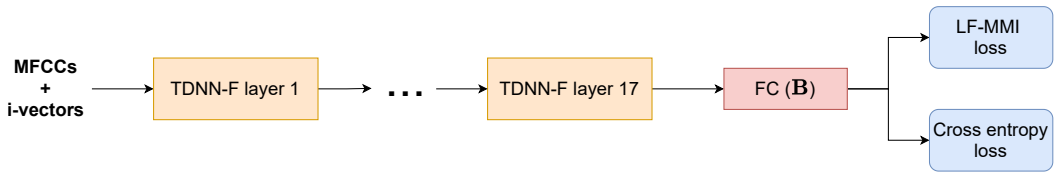


Fig. 2.7 Network architecture for ASR acoustic modeling composed of TDNN-F layers followed by the fully-connected bottleneck layer $\mathbf{B}$ which branches into the computation of the two loss functions, LF-MMI and cross-entropy. The skip connections between TDNN-F layers are not shown for the sake of simplicity.

The acoustic model, used to design and evaluate the anonymization techniques in and after Chapter 5, is a deep neural network which takes MFCCs appended with i-vectors for speaker adaptation [251]. It is composed of 17 TDNN-F layers[4] having 1536 neurons in the hidden layer and 160 neurons in the linear bottleneck, followed by a 256-dimensional fully connected layer $\mathbf{B}$ which leads to two branches that compute the LF-MMI loss and the cross-entropy loss over the tied states as shown in Figure 2.7. This network needs alignment between the observations and the HMM state sequence before starting the training, which is obtained using an HMM that is trained using the iterative Baum-Welch algorithm [235] to marginalize over all possible state sequences that could have generated the observations. At test time, the Viterbi algorithm [89] is used to find the most likely sequence of HMM states (S) that emit the sequence of acoustic observations $\mathbf{O}$, and thereby also give the likelihood of observing $\mathbf{O}$ given the state sequence S. The sequence of acoustic observations is passed through the FST which is the composition of the acoustic model, the context dependency, the lexicon, and the language model. There may be several paths that lead to alternative transcriptions for the same input. Picking the most likely node at each time step (i.e., greedy search) may not lead to the best transcription, hence multiple best paths are considered together at each time step and the remaining less likely paths are pruned. This ensures that the most likely path emerges as the winner and is referred to as the beam search algorithm. The number of paths stored at each time step is called the beam width. It is widely used in ASR and machine translation where the most likely output sequence could not be found using the greedy approach (i.e., beam width is equal to 1). A larger beam width ensures better results but requires the storage of more alternate transcriptions thereby increases the computational cost.

The conventional HMM-DNN approach is quite effective and widely used for ASR, but not without its limitations [100]. The major criticism for this approach is the complexity of its pipeline and the requirement for human expertise. A pretrained GMM-HMM model is needed to generate triphone states that are used as the training targets for the cross-entropy branch of the DNN acoustic model. Separately prepared language model, lexicon and acoustic model are glued together which might compound the overall errors of the ASR system. Moreover, a lexicon must be prepared by expert linguistic rules which might not be available for low-resource languages. The end-to-end approach provides a solution to these issues by subsuming the different models into a single neural network. It has been shown that they perform reasonably well as compared to the conventional pipeline [281], and that they are well suited for low-resource settings [322, 259].

End-to-end approach As a holistic solution to these limitations, recent years have seen rapid development in the domain of end-to-end speech recognition which aims to directly transcribe graphemes (i.e., lexical

[4] TDNN-F layers are described in Figure 2.5.

characters) from speech instead of phonemes, thereby collapsing all the components of the conventional pipeline into a single neural network which is trained in an end-to-end fashion. Ideally, the end-to-end ASR network optimizes its parameters directly based on the sequence-level transcription accuracy, which is the true measure of ASR performance. In practice, a language model is used to re-score the outputs produced by the ASR network which helps them to achieve competitive performance compared to the conventional pipeline [100]. It is reasonable to use a language model because they are trained on additional text-only data that provides realistic prior distribution over words and corrects the mistakes made by the end-to-end network.

Replacing the composite pipeline of conventional ASR with a single neural network requires innovation, both in terms of the training objective as well as the architecture. Graves et al. [100] proposed to use connectionist temporal classification (CTC) as the training objective for an end-to-end ASR network containing five layers of BLSTM with 500 cells each. It does not require a pre-defined alignment between the acoustic features $\{\mathbf{O}_i\}_{i=1}^{N}$ and the corresponding true grapheme label sequence $\{Y_i\}_{i=1}^{N}$, where $Y_i = [y_1, \ldots, y_c, \ldots, y_C], y_c \in \mathcal{G}$ with $\mathcal{G}$ being the set all grapheme symbols including characters, punctutations and space, and C is the number of characters in the transcription of the i-th utterance. Instead, it uses the conditional distribution $P(\mathbf{a}|\mathbf{O})$ which gives the probability of a possible alignment sequence $\mathbf{a} = [\bar{y}_1, \ldots, \bar{y}_T]$ given the acoustic observation sequence $\mathbf{O}$ of length T frames, where $\bar{y}_t \in \mathcal{G}$. The characters $\bar{y}_t$ in a given alignment sequence are obtained by repeating y_c to match the length of $\mathbf{O}$. The probability of an alignment can be obtained using the chain rule and it is simplified by a conditional independence assumption:

$$P(\mathbf{a}|\mathbf{O}) = \prod_{t=1}^{T} P(\bar{y}_t|\bar{y}_1, \ldots, \bar{y}_{t-1}, \mathbf{O}) \approx \prod_{t=1}^{T} P(\bar{y}_t|\mathbf{O}). \tag{2.19}$$

We must marginalize over all possible alignment sequences $\mathbf{a}$ to get the probability of the grapheme sequence G_i but in practice, it is not feasible to sample all possible alignments, so Monte-Carlo sampling is used to compute the CTC loss and its gradient. The final loss function $\mathcal{L}_{\text{ctc}}$ is as follows:

$$\mathcal{L}_{\text{ctc}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \log P\left(\mathbf{a}_j^{(i)} \middle| \mathbf{O}_i\right). \tag{2.20}$$

Here $\mathbf{a}_j^{(i)}$ is one of the M possible alignment sequences for i-th utterance, sampled using Monte-Carlo.

Attention-based approaches are an alternative to CTC which do not make any conditional independence assumptions. Instead, they consider all the previous outputs and the whole input sequence to estimate the posterior [16, 319]. The attention mechanism does not require an intermediate alignment representation, hence the loss is computed as follows:

$$\mathcal{L}_{\text{att}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \log P(y_c|y_1, \ldots, y_{c-1}, \mathbf{O}_i). \tag{2.21}$$

Although several different architectures based on recurrent [111, 20, 203] and convolutional [342] neural networks were proposed for end-to-end ASR, Watanabe et al. [319] proposed to combine CTC and attention-based mechanisms through multi-objective training, which is used in Chapters 3 and 4 to design and evaluate speaker anonymization techniques. As shown in Figure 2.8, it follows the so-called encoder-decoder architecture where the encoder, composed of four BLSTM layers with 320 units each, transforms the input sequence into a new bottleneck representation $\mathbf{B}$, and the decoder, with a single unidirectional LSTM layer having 320 units, predicts the grapheme sequence from $\mathbf{B}$. The attention layer

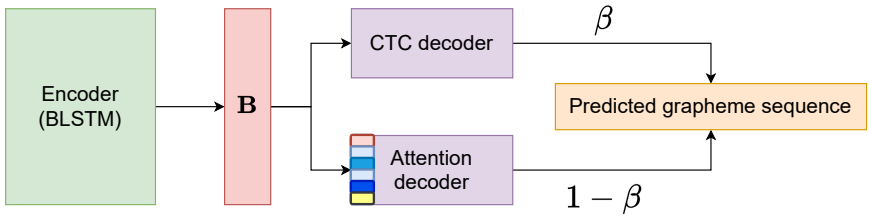


Fig. 2.8 End-to-end ASR architecture with multi-objective training consisting of CTC and attention-based loss functions combined by a hyperparameter β, where $0 < \beta < 1$. The attention layer is shown as the heatmap in the attention decoder which assigns combination weights to the bottleneck representation $\mathbf{B}$ before processing it by the LSTM layer.

sits in between the encoder and the decoder and tells the decoder at each time step how much weight should be assigned to a particular part of $\mathbf{B}$ to predict the output grapheme at that time step with the least amount of error. The weights of the attention layer are learned within the end-to-end framework [147].

2.2.4 Speech synthesis

As opposed to ASR, the goal of speech synthesis also known as text-to-speech (TTS) is to convert a text string into a speech waveform. We have already seen how there have been historical efforts to produce speech sounds either using a physical model like von Kempelen's "speaking machine" or electronic models like Dudley's VODER. Modern-day TTS technology has greatly advanced especially with the introduction of statistical models, like neural networks. The current state-of-the-art TTS models can produce almost natural-sounding speech from any text, in several voices, and multiple languages. The progress in modern TTS technology can be divided into three generations of systems [216], namely unit selection, statistical parametric speech synthesis (SPSS), and neural speech synthesis. The exact formulation of these three systems is out of scope for this thesis, but they are all based on some essential fundamental principles which will be briefly covered here. Finally, a small note about the evaluation of TTS systems is mentioned at the end of this section.

Figure 2.9 shows a general schematic diagram for TTS systems. It is composed of the frontend, the acoustic model, and the waveform generator. Note that unit selection methods directly generate a waveform using the output of the frontend. In contrast, SPSS methods employ an acoustic model to first convert the frontend's output into spectral parameters of the target speech and then generate the target speech using a waveform model. We briefly describe these three blocks of TTS systems below.

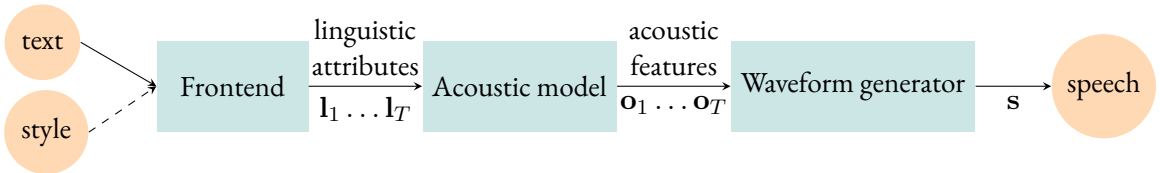


Fig. 2.9 General schema of a TTS system with the style features as optional input to the frontend.

The frontend The first challenge is to predict non-linguistic attributes such as speaking style, emotions, and prosodical cues simply from the plain text without any auxiliary information. Speaker traits, which are generally derived from the fixed number of speakers in the training data set and are otherwise impossible to

predict simply from text, are also needed as input to the frontend. The frontend contains a large number of meticulous rules hand-crafted by linguists for different languages which convert written form of words to their spoken form by normalizing them into tokens found in a dictionary, assigning them a part-of-speech class, getting the exact sequence of phonemes to pronounce, as well as predicting the intonation pattern and phrase breaks. It is costly to build and maintain the frontend for different languages since it requires careful analysis of phonetic inventory and different ways to pronounce certain complicated tokens like abbreviations, numbers, currency symbols, etc. Some end-to-end TTS systems [314, 11] aim to replace this complex pipeline with a neural network that encodes linguistic attributes in its hidden layers and requires only the sequence of letters as input, but they are still in their infancy and make some crucial pronunciation mistakes [258] which prevent their commercial use. There have also been efforts to capture general speaking style including speaking rate, emotions, prosodical patterns using neural network embeddings [315]. Given the input text and the style embedding, expressive speech can be synthesized. Moreover, intonation and pronunciation mistakes can be corrected by using morphological features [287] that are present in the text. Nevertheless, all commercial TTS systems still maintain the traditional frontend because it can be easily understood and corrected when it makes mistakes.

At this point, we know that it is hard to reproduce the exact linguistic and paralinguistic attributes that were present in the original speech simply from the text content. Moreover, the generated utterance may not be as diverse as the natural speech due to limited number of speakers in the training data for TTS. Therefore, given these drawbacks, we discard the seemingly simple solution to achieve the privacy objective, i.e. to convert speech to text using ASR and then synthesize speech from this text using TTS, due to the destruction of usable information[5] which is contrary to the goals of this thesis. Having addressed this possibility, we move on to describe the remaining components in the TTS pipeline that aim to produce a waveform using the linguistic and non-linguistic features generated by the frontend.

Acoustic model The TTS acoustic model converts linguistic features generated by the frontend into a sequence of acoustic features, such as a magnitude spectrogram, through a regression task that can be easily solved using a neural network. One popular approach is to use an autoregressive[6] neural network-based acoustic model, such as the one proposed by Lorenzo-Trueba et al. [184], to generate the Mel-spectrogram $\mathbf{O}$ of an utterance given the linguistic features $[\mathbf{l}_1, \ldots, \mathbf{l}_T]$ of T frames extracted from the text. This approach generates the Mel-spectrogram corresponding to the target speaker.

The autoregressive acoustic model, as shown in Figure 2.10, is a sequence model with feed-forward and recurrent LSTM layers, where the ouput acoustic feature at time t is produced depending on the whole input sequence and some of the acoustic features at previous times. Hence, the probability of observing output acoustic features is defined as follows:

$$P(\mathbf{o}_1, \ldots, \mathbf{o}_T | \mathbf{l}_1, \ldots, \mathbf{l}_T) = \prod_{t=1}^{T} P(\mathbf{o}_t | \mathbf{o}_{t-T'}, \ldots, \mathbf{o}_{t-1}, \mathbf{l}_1, \ldots, \mathbf{l}_T). \tag{2.22}$$

The acoustic model generates $\mathbf{o}_t$ based on the previous T' outputs and the whole sequence of input linguistic features.

Waveform generation The final component of the TTS pipeline is the waveform generator, which processes the acoustic features to produce an intelligible waveform. Unit selection speech synthesis bypasses the acoustic modeling and instead, generates the waveform using a concatenation approach. It assumes

[5] Transcription errors introduced by the ASR

[6] Autoregressive model relies on its past outcomes to predict the current one as in Equation (2.22).

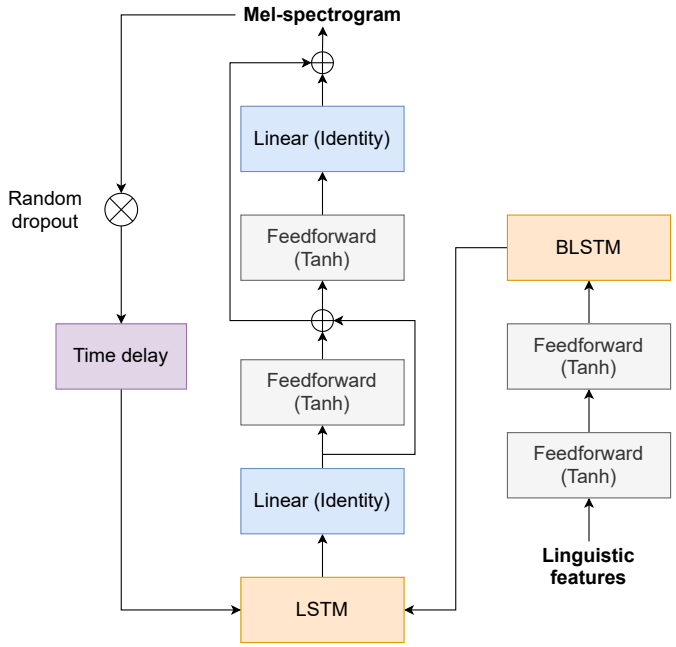


Fig. 2.10 Autoregressive network architecture for the TTS acoustic model [184]. The feedforward layers have Tanh activation and the linear layers have identity activation. It has skip connections to reinforce the noisy signal, and the time delay block passes the current output Mel-spectrogram back to the LSTM layer with a random dropout.

that a large data set of natural speech spoken by real humans already exists, and at synthesis time a plausible sequence of speech segments that satisfy the given linguistic criteria are selected and stitched together to generate the waveform. The units that are stitched together are usually diphones, which are nothing but the second half of one phone and the first half of another to retain the overlap between the two, thereby capturing the coarticulation boundary. Therefore the data set must ideally include multiple instances of all possible diphones in the considered language. This approach used to be popular due to the naturalness of the generated speech. Yet there are several glaring limitations, for example, the concatenation of waveform segments may not be smooth at the joins which may result in perceptual glitches while listening to the produced audio. It is also not possible to have all the possible speaking styles in the data set and it is certainly quite cumbersome to scale this approach to include new speakers.

SPSS methods try to alleviate the abovementioned limitations of unit selection by estimating the acoustic parameters of the target speech, instead of using pre-recorded samples. The acoustic parameters can then be manipulated or used directly to generate the waveform with desired properties. The task of waveform generation just using the acoustic features is still challenging because they are composed of logmel features only. These features, derived from the magnitude spectrogram, are not sufficient to reconstruct the original signal since we also require the exact phase of each sine wave corresponding to that particular speech sound. The waveform generator, also called the vocoder, needs to predict this missing information for producing an intelligible speech signal. Traditional vocoders like STRAIGHT [149] or WORLD [206] provide various analysis algorithms to be applied on the speech signal to efficiently extract the spectra, the fundamental frequency, and the aperiodicity from all frames. They also provide synthesis algorithms that can combine this information and produce a good quality speech waveform in real-time, but they ignore phase prediction

and instead make a minimum phase assumption, which leads to noticeable speech distortion in low F0 regions [206].

Autoregressive neural networks, such as Wavenet [303] and SampleRNN [199], have shown promising results as waveform generators, but they are highly inefficient and hard to parallelize due their sequential generation process. This limitation is relieved by Neural source-filter (NSF) models that are waveform generators [313] inspired by the classical source-filter paradigm of speech synthesis [114, 198]. The traditional source-filter model aims to mimic the speech production mechanism by assuming a *source* of sound that produces a discrete-time excitation signal $\bar{e}$, and a *filter* which modulates the frequency components of the excitation signal to convert it into a phoneme-like speech signal $\mathbf{s}$. The source can produce either a periodic signal, such as an impulse train or a sine wave, to mimic the voiced excitation that contributes to the inherent harmonic structure in natural speech, or a noise signal for the unvoiced turbulence across the vocal tract. The filter acts like the vocal tract which produces formants in the spectral envelope due to its characteristic shape, and therefore must be individually designed for each type of sound. In its simplest form, it is nothing but the linear transformation of the excitation signal and the previous output, and the output at each time instance is specified by the following equation:

$$\mathbf{s}[n] = \bar{e}[n] + \sum_{k=1}^{t} c_k \cdot \mathbf{s}[n-k]. \tag{2.23}$$

Here, c_k are the filter coefficients that may vary for different speech sounds, and the output is produced at each time instance by considering t previous outputs, which is also known as the order of the filter.

The NSF model[7] shown in Figure 2.11 is much more advanced than the simple, linear source-filter formulation. First, it contains a condition module which takes $[\mathbf{c}_1, \ldots, \mathbf{c}_T]$ as input, where $\mathbf{c}_t = [p_t, \mathbf{o}_t]^\top$ is composed of the fundamental frequency p_t and the acoustic feature $\mathbf{o}_t$ for the t-th time frame. It upsamples the fundamental frequency and outputs $[p'_1, \ldots, p'_{N_\mathbf{s}}]$ to match the length of the target waveform. It also processes the acoustic features $\mathbf{o}_t$ using BLSTM and convolutional layers, and concatenates the output with the upsampled fundamental frequency to get the condition feature sequence $[\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{N_\mathbf{s}}]$.

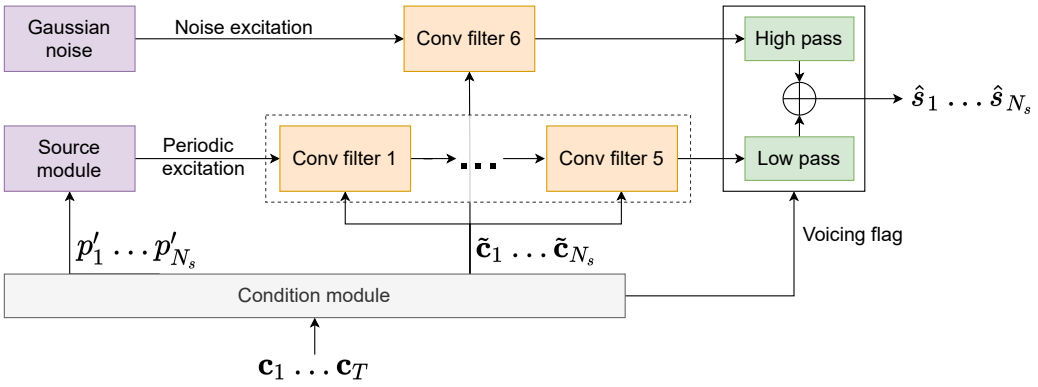


Fig. 2.11 NSF model architecture.

Next, the source module which accepts $[p'_1, \ldots, p'_{N_\mathbf{s}}]$ as input and generates a periodic signal for voiced sounds (i.e., a mixture of sine waves parametrized by their amplitude and phase) as an excitation based on the value of p'_n. Another module generates a separate noise excitation for unvoiced sounds. The periodic

[7] We describe here the state-of-the-art NSF model which is referred to as Harmonic-plus-Noise NSF model in [313].

signal, combined with the condition features $[\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{N_\mathbf{s}}]$, is transformed through a series of *Conv filters* which contain multiple dilated convolutional layers with residual connections, while the noise excitation is transformed using a single Conv filter. The periodic signal is subjected to a lowpass filter to preserve dominant regions in higher frequencies, while the noise excitation is passed through a highpass filter to preserve lower frequencies. There are two configuration of the bandpass filters depending on the value of the voicing flag. They are configured such that the higher frequencies are preserved for voiced regions, while the lower frequencies for unvoiced regions. The output of the two bandpass filters is summed up to get the estimated target waveform $[\hat{s}_1, \ldots, \hat{s}_{N_\mathbf{s}}]$.

The Conv filters are learned to minimize the log spectral amplitude distance:

$$\mathcal{L}_{\text{NSF}} = \frac{1}{2TL} \sum_{t=1}^{T} \sum_{k=1}^{L} \left[\log \frac{|\mathcal{F}_t[k]|^2}{\left|\hat{\mathcal{F}}_t[k]\right|^2} \right]^2 . \tag{2.24}$$

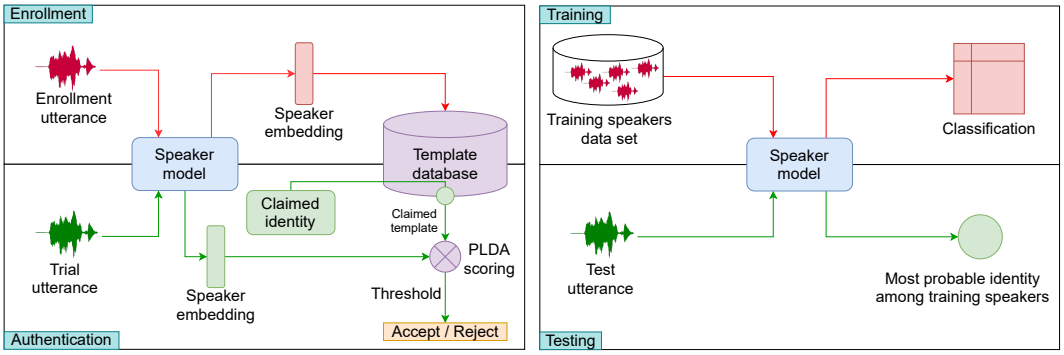
Here, $\mathcal{F}_t[k]$ and $\hat{\mathcal{F}}_t[k]$ denote the k-th short-time Fourier transform coefficient of the t-th time frame obtained from the original and the predicted waveform, respectively.

Evaluation The evaluation of output speech is usually performed using mean opinion scores (MOS) obtained using subjective listening tests by human subjects [134]. Several such subjects with different gender and age profiles rate the speech on a perceptual scale based on its quality, intelligibility, and naturalness. This method of evaluation is costly, labour intensive, and slow, hence in this thesis we objectively evaluate the generated speech samples using ASR systems, which may not be a very effective measure of qualitative attributes of speech but highly correlate with human intelligibility [14, 88, 107].

A flexible high-quality TTS system that generates personalized utterances of a given target speaker can be used to clone the identity of any arbitrary person. But these threats are not investigated in this work and the solutions to such issues are beyond the scope of this thesis.

2.2.5 Automatic speaker recognition

Automatic speaker recognition is the task of recognizing the speaker of a given speech utterance. As mentioned in Section 1.1, speaker information in the speech signal is quite sensitive since it describes several attributes related to the speaker's identity and personality. Campbell, in his seminal tutorial on speaker identification [34], lists several factors responsible for the speaker-dependent characteristics present in speech signal due to speech production mechanism. Most of the factors arise due to the physiology and the shape of the vocal tract of the speaker. When the acoustic wave passes through the vocal tract, its frequency is modulated by the dominant resonances (i.e., formants), which can be easily observed in the spectral envelope of the signal. There are other speaker-dependent factors that emerge due to the source of excitation, generated by lungs and are then carried across the trachea over to the vocal folds. The source is responsible for phonetic features such as voicing, frication, whisper, etc. along with the fundamental frequency F0. The mass and length of the vocal folds are the defining properties for the fundamental frequency, hence it is a speaker- as well as gender-dependent characteristic. Some other characteristics that describe the style of speaking like the speaking rate, the dialectal shift in frequencies for speakers of similar language, and general prosodic patterns that emerge from conversations with specific vocabulary such as technical or professional settings, can also contribute to the speaker's identity.



(a) Automatic Speaker Verification (ASV) (b) Automatic Speaker Identification (ASI)

Fig. 2.12 Automatic speaker verification vs. automatic speaker identification. Red arrows indicate the enrollment or training flow, and green arrows indicate the authentication or testing flow. Note that the speaker model trained to classify speakers in the case of ASI can also be used to extract speaker embeddings for ASV, just like x-vectors.

Speaker recognition technologies are largely deployed in forensic studies [7, 289, 8] and telephone banking systems.[8,9,10] The techniques for automatic speaker recognition can be categorized into automatic speaker verification (ASV), i.e., authenticating the identity claimed by the speaker, and automatic speaker identification (ASI), i.e., determining the identity within a set of known speakers [26]. The schematic diagram of these two methods is depicted in Figure 2.12. ASV (Figure 2.12(a)) comprises two successive phases: enrollment and authentication. In the former, speakers are enrolled using discriminative speaker embeddings which are extracted from enrollment utterances. Speaker embeddings must have some characteristic features [154]. They must have large between-speaker variability and small within-speaker variability to get similar embeddings for different utterances from the same speaker. They must be easy to extract and difficult to impersonate, as well as robust against noise and distortion. The most popular embeddings called x-vectors [269] are obtained from an intermediate layer of a neural network trained to perform speaker classification. In the latter phase, the x-vector $\mathbf{v}_t$ extracted from the utterance of an unknown speaker (called trial utterance) is compared with the x-vector $\mathbf{v}_e$ of the speaker whose identity is being claimed, and a log-likelihood ratio score is computed by probabilistic linear discriminant analysis (PLDA) [150]. To compute the PLDA score, it is assumed that the speaker embedding $\mathbf{v}$ is generated from a linear Gaussian model as $p(\mathbf{v}|y,z) = \mathcal{N}(\mathbf{v}|\mu_s, Vy + Dz + R)$, where μ_s is the global mean in the speaker space, the columns of V capture speaker variability (eigenvoices) with y depending only on the speaker, the columns of D encode channel variability (eigenchannels) with z varying from one recording to another, and R is the diagonal matrix of residual variances. A PLDA model is essentially a Gaussian distribution in the speaker embedding space given by the marginal density

$$p(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mu_s, \Phi_b + \Phi_w), \tag{2.25}$$

[8] https://www.us.hsbc.com/customer-service/voice/

[9] https://www.chase.com/personal/voice-biometrics

[10] https://www.lloydsbank.com/contact-us/voice-id.html

Table 2.1 Original network architecture for the x-vector speaker classification model as presented in [269, Table 1].

Layer name	Layer type	Layer context	Dilation	Total context	input×output
frame1	TDNN	$[t-2, t+2]$	No	5	120×512
frame2	TDNN	$\{t-2, t, t+2\}$	Yes	9	1536×512
frame3	TDNN	$\{t-3, t, t+3\}$	Yes	15	1536×512
frame4	TDNN	$\{t\}$	No	15	512×512
frame5	TDNN	$\{t\}$	No	15	512×1500
stats pooling	Linear	$[0, T)$	NA	T	1500T×3000
segment6	Linear	$\{0\}$	NA	T	3000×512
segment7	Linear	$\{0\}$	NA	T	512×512
softmax	Output	$\{0\}$	NA	T	512×N

where $\Phi_b = VV^\top$ and $\Phi_w = DD^\top + M$ are the between-class and within-class covariance matrices, respectively. The similarity score between the two x-vectors is computed as

$$l_{\text{PLDA}}(\mathbf{v}_t, \mathbf{v}_e) = \frac{p(\mathbf{v}_t, \mathbf{v}_e)}{p(\mathbf{v}_t)p(\mathbf{v}_e)}. \tag{2.26}$$

The denominator term of Eq. (2.26) can be computed using Eq. (2.25), while the numerator is computed using

$$p(\mathbf{v}_t, \mathbf{v}_e) = \mathcal{N}\left(\begin{bmatrix}\mathbf{v}_t \\ \mathbf{v}_e\end{bmatrix} \middle| \begin{bmatrix}\mu_s \\ \mu_s\end{bmatrix}, \begin{bmatrix}\mathbf{C} & \Phi_b \\ \Phi_b & \mathbf{C}\end{bmatrix}\right), \tag{2.27}$$

where $\mathbf{C} = \Phi_b + \Phi_w$ is the total covariance matrix.

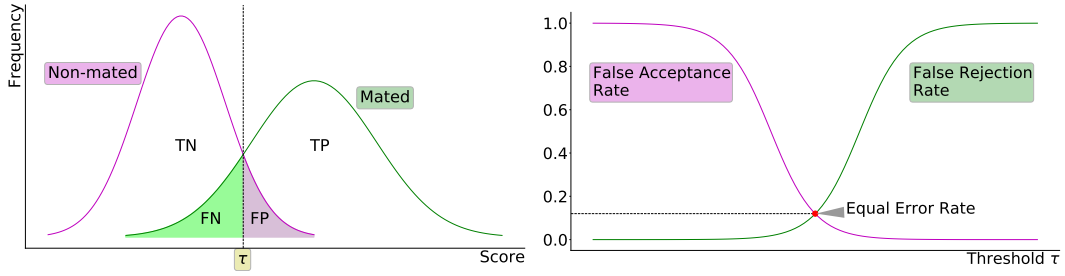
After obtaining the score l_{PLDA}, the ASV system decides whether the trial utterance is from the considered enrollment speaker or not by comparing the score with a threshold. As opposed to the open-set (rejection/acceptance) ASV task, ASI (Figure 2.12(b)) is a closed-set task in which a speaker classifier (e.g., similar to the one used to obtain x-vectors) is trained on training utterances from multiple speakers to later classify the identity of each test utterance as one of the known training identities.

X-vectors [269], which formulate ASI as a sequence classification task, have made the training and evaluation of an efficient ASI model quite straightforward. The architecture of the neural network used for extracting x-vectors is presented in Table 2.1, where the input is a sequence of speech features extracted from the utterance of a speaker, such as MFCCs, and the output is the posterior over the speaker classes. This task is accomplished using five TDNN layers followed by a statistical pooling layer and a fully connected classifier. Since the speaker information is present throughout the utterance, the statistical pooling layer computes the mean and standard deviation of the feature sequence produced by the preceding TDNN layers to retain the global speaker-related characteristics and diminish the local linguistic variations in speech. The output of intermediate layer just after statistical pooling (i.e., *segment6*), being rich in speaker information, is used as the x-vector. The design and evaluation of the ASV system are not so trivial as it requires the speaker embedding to capture relevant speaker information which generalizes to unseen speakers. The scoring mechanism must also produce values that truly discriminate closer speakers from farther ones. And finally, the selection of the decision threshold is critical for the performance of the ASV system. It must be carefully calibrated based on the application, for example speaker authentication for bank transactions requires a

stringent threshold to avoid frauds while mobile phone unlocking requires a liberal threshold to allow quick access to users.

The ASI model is conventionally evaluated using performance metrics, such as accuracy, which measure how frequently the model outputs the correct class. On the contrary, ASV systems are evaluated based on the errors they make. Suppose there are N trials, i.e., pairs of enrollment and trial utterances in a data set and each trial is either *genuine* (mated) or *impostor* (non-mated). Genuine, here referred as positive, trial represents the case when the speaker is really who he/she claims to be, hence the enrollment and the trial utterances belong to the same speaker. Contrary to that, in case of impostor or negative trial, the enrollment and the trial utterances belong to different speakers. If the model outputs 'positive' for TP examples which are truly positive, 'positive' for FP examples which are negative, 'negative' for FN examples which are positive, and 'negative' for TN examples which are truly negative, then the accuracy is given by: $\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$, where $\text{TP} + \text{TN} + \text{FP} + \text{FN} = N$. It can be computed by simply counting the instances corresponding to TP, FP, TN, and FN, but the efficacy of ASV authentication is dependent upon two types of errors which are false acceptance rate (FAR), also known as type-I error and is given by $\frac{\text{FP}}{\text{FP}+\text{TN}}$, and false rejection rate (FRR), also known as type-II error and is given by $\frac{\text{FN}}{\text{FN}+\text{TP}}$. FAR and FRR are in turn dependent on the selected decision threshold, τ.

Figure 2.13(a) shows the mated and non-mated score distributions. All biometric authentication systems must choose a threshold τ by observing these two distributions such that the values of FN and FP are minimized. It would be ideal if there were no overlap between mated and non-mated score distributions, but in practice, there is always some overlap due to the diversity of enrolled persons, imperfect scoring mechanisms, oversimplified modeling assumptions and limited training data. Hence, there is always a tradeoff between FAR and FRR as observed in Figure 2.13(b). Authentication systems may choose the threshold based on the sensitivity of their applications, but to evaluate them a fixed point is often chosen where the FAR is equal to the FRR. The error at this point is called the Equal Error Rate (EER). A lower EER indicates a better ASV system.



(a) Mated and non-mated score distributions

(b) Error curves as functions of the threshold

Fig. 2.13 ASV Score distribution and threshold.

Implications for the assessment of privacy ASV and ASI systems are extensively used to design and simulate privacy attackers for the evaluation of the techniques proposed in this thesis as described in the next chapter. It is imperative to understand how to interpret the obtained results when the level of privacy protection is measured under strong hostile conditions. This thesis endeavors to show that it is possible to fine-tune the degree of hostility or the strength of the malicious entity who is trying to re-identify protected speakers. Hence, the anonymization techniques are evaluated based on their resilience against such strong criteria. Broadly speaking, anonymization does not imply that the features conveying speaker information are *completely* deleted from the speech signal or that it is even possible to do so. Instead, it means that the

confusion for the attacker has been significantly increased by transforming the features of a given speaker such that they are indistinguishable from the other speakers. In that case, the degree of privacy is contingent upon the speaker discrimination capacity possessed by the attacker after anonymization and the set of enrollment speakers who are shortlisted as the possible targets. To that end, Section 3.1 gives a detailed guideline for designing the best possible attackers so that the degree of protection is truly measured.

The EER is widely used to evaluate biometric authentication systems, and a higher EER may indicate the inadequacy of the attacker who is trying to infer the true identity of the speaker. But some properties of EER may have limiting implications for it to be used as a general measure of privacy protection and evaluate the strength of a vast range of attackers. First, it assigns the same cost to false alarms (FP) and misses (FN) which may not be an optimal assumption to model the attacker. The attacker may choose a more relaxed setting to shortlist all possible speaker identities by lowering the threshold, which calls for a generalized evaluation where all possible priors over error costs are considered. Second, it only considers a single point of overlap between the mated and non-mated distributions, whereas it is possible that they are not monotonic and overlap at several points. The scores at the points of overlap may be present on either side of the EER and can be leveraged by the attacker to strengthen the attack. Finally, the EER indicates the overall efficacy of the biometric authentication system in the presence of several enrolled speakers while the actual level of protection for a particular speaker may slightly differ from the EER based on the indistinguishability of their individual score distribution from the overall scores.

In this thesis, some of the abovementioned limitations are alleviated by reporting other suitable metrics of privacy that are described and compared in the next chapter (Section 3.4). Later, some analysis is also provided to measure the best and worst-case privacy protection using re-identification metrics and formal methods such as differential privacy.

2.3 Techniques to transform speaker information

In this section, we review the fundamentals of the three most relevant techniques, i.e., adversarial learning, speech transformation, and voice conversion, that are widely used by researchers to modify speaker information in speech data. The remaining chapters of this thesis employ these techniques for their potential to hide/remove speaker-related biometric information from speech.

2.3.1 Adversarial learning for speech

Domain adversarial training helps to adapt neural network classifiers to a new domain without requiring labeled data in the new domain. The original paper [92] shows promising results on a handwritten digit classification task where the features learned using domain adversarial training are distributed identically, whether the image is grayscale or RGB (colored). It has been extensively applied to speech data since its inception. The idea of adversarial training enables neural networks to learn intermediate features in the form of hidden layers that are indiscriminate towards data belonging to similar classes but originating from different domains. For instance, the intermediate features for a particular phoneme class learned using domain adversarial training will be distributed almost identically, whether it is spoken in different ambient noise or by different speakers. Domain adversarial training is implemented as a neural network architecture with a so-called *adversarial branch* that predicts the domain, and a special layer just before this branch called the *gradient reversal layer* which scales the gradients that are being backpropagated through this layer using a negative scalar value. This layer ensures that the parameters of the preceding network are shifted in such a way that they implicitly remove the domain information from the representation, thereby making it invariant to irrelevant factors of variation that do not contribute towards the main task. This property of domain adversarial training is quite appealing to the speech community because speech signals are very

expressive and full of factors of variation such as speaker's identity, channel information, emotions, language, accents, etc., but generally, machine learning models are trained to classify or predict only a single attribute from data, hence it is desirable to get rid of irrelevant attributes.

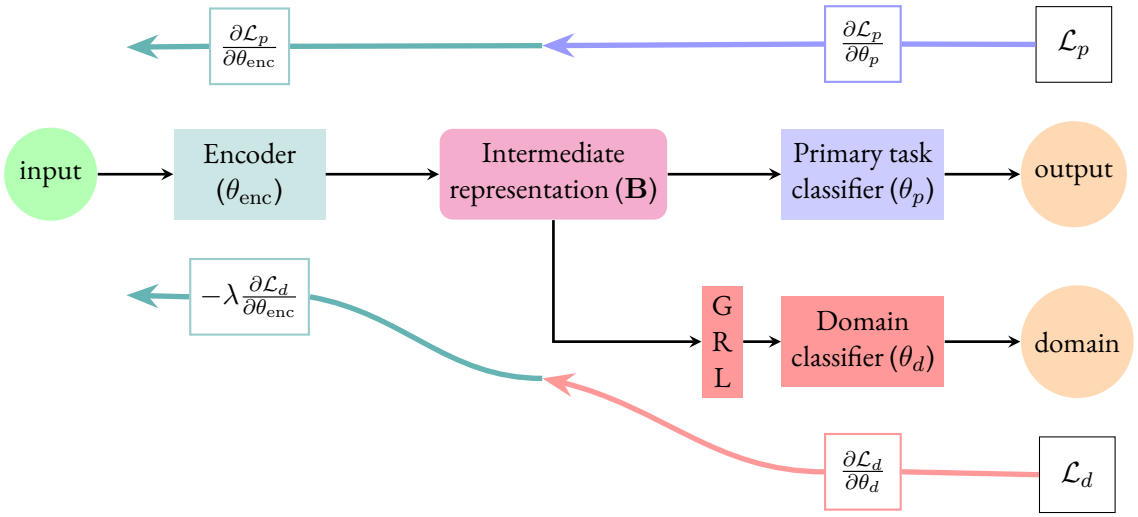


Fig. 2.14 General architecture for domain adversarial training of neural networks. Black arrows indicate forward propagation; purple, teal and red arrows indicate backpropagation of gradients for the primary task classifier, the encoder and the adversarial branch, respectively. The red GRL block refers to the gradient reversal layer with λ as the gradient reversal coefficient.

Speech researchers have employed adversarial training not only for domain adaptation [297, 56], but for enhancing speech quality [201, 178], training noise-robust ASR [260, 271], and learning representations which are invariant towards speaker [5, 200, 299], language [327, 4] and accents [278]. Figure 2.14 shows a general architecture for domain adversarial training which takes input speech features and transforms them into an intermediate representation ($\mathbf{B}$) using an encoder neural network with parameters θ_{enc}. The intermediate representation is fed to the primary task classifier with parameters θ_p which estimates the desired output, such as the transcription, emotional valence, etc., and computes the loss $\mathcal{L}_p$. In parallel, $\mathbf{B}$ is also fed through the gradient reversal layer, which leaves it unchanged during the forward propagation and passes it to the adversarial branch, also called the domain classifier with parameters θ_d, which predicts the domain label and computes the adversarial loss $\mathcal{L}_d$. The parameters θ_{enc}, θ_p, and θ_d are jointly estimated by solving the following minimax optimization problem:

$$\min_{\theta_{\text{enc}},\theta_p} \max_{\theta_d} \mathcal{L}_o(\theta_{\text{enc}}, \theta_p, \theta_d). \tag{2.28}$$

Here, $\mathcal{L}_o$ is the overall loss given by: $\mathcal{L}_o(\theta_{\text{enc}}, \theta_p, \theta_d) = \mathcal{L}_p(\theta_{\text{enc}}, \theta_p) - \lambda \mathcal{L}_d(\theta_{\text{enc}}, \theta_d)$, and λ is the gradient reversal coefficient which decides the trade-off between the primary and the adversarial objectives. A higher λ increases robustness of $\mathbf{B}$ towards the domain but may decrease its efficacy towards the primary task.

During the backward pass, the parameters of the primary task classifier and the domain classifier are updated according to their respective losses $\mathcal{L}_p$ and $\mathcal{L}_d$, but the encoder's parameters are updated with respect to both losses as opposing goals. The goal of domain adversarial training is to make $\mathbf{B}$ invariant towards the domain label, hence the encoder parameters must be shifted in the direction such that the new $\mathbf{B}$ maximizes $\mathcal{L}_d$, while minimizing $\mathcal{L}_p$ at the same time. The gradient descent update rules for each part of

the network are as follows:

$$\theta_p \leftarrow \theta_p - \eta \frac{\partial \mathcal{L}_p}{\partial \theta_p}, \tag{2.29}$$

$$\theta_d \leftarrow \theta_d - \eta \frac{\partial \mathcal{L}_d}{\partial \theta_d}, \tag{2.30}$$

$$\theta_{\text{enc}} \leftarrow \theta_{\text{enc}} - \eta \left(\frac{\partial \mathcal{L}_p}{\partial \theta_{\text{enc}}} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_{\text{enc}}} \right), \tag{2.31}$$

where η is the learning rate hyperparameter.

In the context of this thesis, we will investigate whether speaker-related information can be removed using domain adversarial training. Meng et al. [200] train an ASR acoustic model with an additional speaker adversarial branch and show that while it improves the performance of ASR, the intermediate features from the same phoneme belonging to different speakers are qualitatively more identically distributed after using adversarial training. Adi et al. [5] observe that the accuracy of speaker classification performed using the intermediate representation of an end-to-end ASR network is significantly reduced after training it with speaker adversarial branch. Tu et al. [299] also show an increase in emotion recognition when the classifier is trained with a speaker adversarial branch, and the effect of the speaker becomes negligible over the performance of the network. These studies demonstrate the potential of domain adversarial training for designing a privacy-preserving mechanism to identify and remove speaker-related information from speech. Chapter 4 presents our investigation in this direction.

2.3.2 Speech transformation

Speech transformation, also called voice transformation [276], is a general modification of speech that aims to shift the perceivable physical attributes of an utterance in a certain direction while leaving the linguistic content unchanged. It is often used as a complementary step after speech synthesis to make the output sound more natural through careful rule-based manipulations of the signal. Speech transformation systems are widely used in speech toolkits due to their efficient real-time nature and general applicability. Some of the interesting applications of speech transformation algorithms are emotion simulation in synthetic speech [33] and conversion of speech to sound like songs [58].

Speech transformation algorithms ease the manipulation of prosodic features of speech, such as speaking rate, loudness, pitch, stress pattern, and in effect the overall speaking style, which originate from the source part of the vocal tract (i.e., lungs and vocal folds). Although the speaking style is an abstract concept, speech transformation algorithms can be used to map the style of one speaker over the utterance of another speaker. They are typically not designed to achieve a predefined target but a relative shift from the source instead, such as a faster time-scale, a lower pitch, etc. Speech transformation algorithms are also designed to modify the filter characteristics (recall the source-filter model described in Section 2.2.4), which describe the frequency response of the vocal tract. The frequency response can be simply warped in a certain direction by applying, for example, a bilinear function, expressed as $f(\omega, \alpha) = |-j \ln \frac{z-\alpha}{1-\alpha z}|$, where $\omega \in [0, \pi]$ is the normalized frequency, $\alpha \in (-1, 1)$ is the warping factor, and $z = e^{j\omega}$. It is applied to the spectrum with a predefined domain over it [234] to quickly produce perceivably different voices. Figure 2.15 shows the response of the bilinear function for positive and negative values of α. When $\alpha < 0$, the lower frequency region is compressed and the higher region is stretched, while the reverse happens when $\alpha > 0$.

Despite their simplicity, speech transformation algorithms are hard to design because they require a clear understanding of the processes related to speech production and perception. Several systematic studies [274, 81] have been conducted to find the acoustic correlates of phonetic processes. Researchers have closely analyzed the source characteristics of speech and defined the parameters that can be modified to

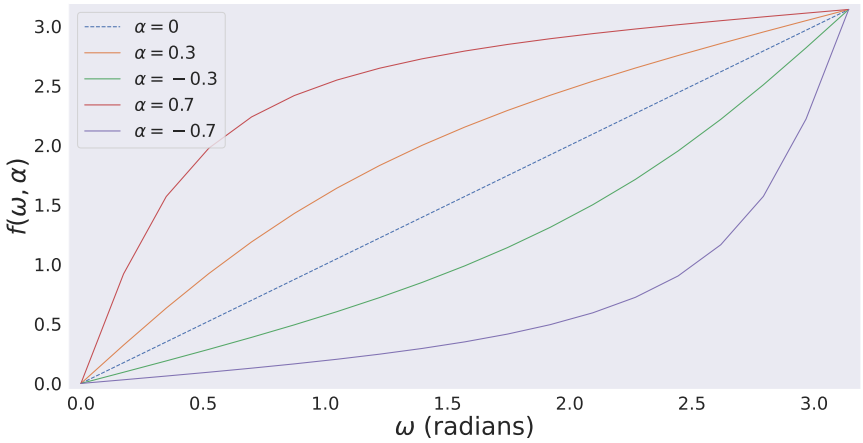


Fig. 2.15 Bilinear function warping the frequency $\omega \in [0, \pi]$ using positive and negative values of $\alpha \in \{-0.7, -0.3, 0, 0.3, 0.7\}$.

perceive certain effects [71, 82, 80]. For example, Stylianou [276] mentions the case that, when someone wants to increase the loudness in a given speech utterance, they must increase the energy of consonant segments rather than vowels because consonants have a short duration yet carry most of the information load in oral communication. This increase in stress also implies an increase in subglottal pressure, thereby an increase in pitch and energy in high-frequency regions. Similarly, increasing the speaking rate also has an effect on pitch. These examples illustrate that the parameters of speech cannot be modified in isolation, and their phonetic and articulatory relationship must be known before implementation, otherwise, the resulting voice loses naturalness.

One of the works that are closely related to this thesis is that of Matrouf et al. [195] who show that speech transformation can be used to transform the voice of impostor speakers such that it closely mimics the mated distribution of the speaker recognition system. The goal of this mechanism is similar to voice spoofing [325] where an impostor is caused to be accepted as a legitimate target speaker by exploiting the vulnerabilities of the speaker recognition system. Such a mechanism can also be used to fool the algorithms used by an attacker to identify the true speaker from a data set by making the mated and non-mated distributions indistinguishable from each other, that is not robustly achieved by the given mechanism in the case when the attacker is aware of the transformation algorithm. Most of the proposed approaches in this thesis attempt to do this in rigorous settings for achieving robust anonymization.

2.3.3 Voice conversion

Voice conversion (VC) [265] aims to convert the voice of a speaker to sound like another, while leaving the linguistic content unchanged. VC has a more specific goal than speech transformation in terms of the precision of the target voice achieved after conversion, thereby it also requires identification and modeling of the exact characteristics that are particular to the source as well as target speaker. We have already described in Section 2.2.5 the different factors which contribute towards the vocal identity of each speaker. They include the overall spectral information, sometimes called *timbre*, and prosodic information, such as pitch, duration, and intensity. VC attempts to learn a mapping from source to target speaker characteristics by modifying these relevant factors. Traditional VC approaches use a vocoder at their core to analyze and synthesize the voice with a feature conversion module to map the source to the target speaker.

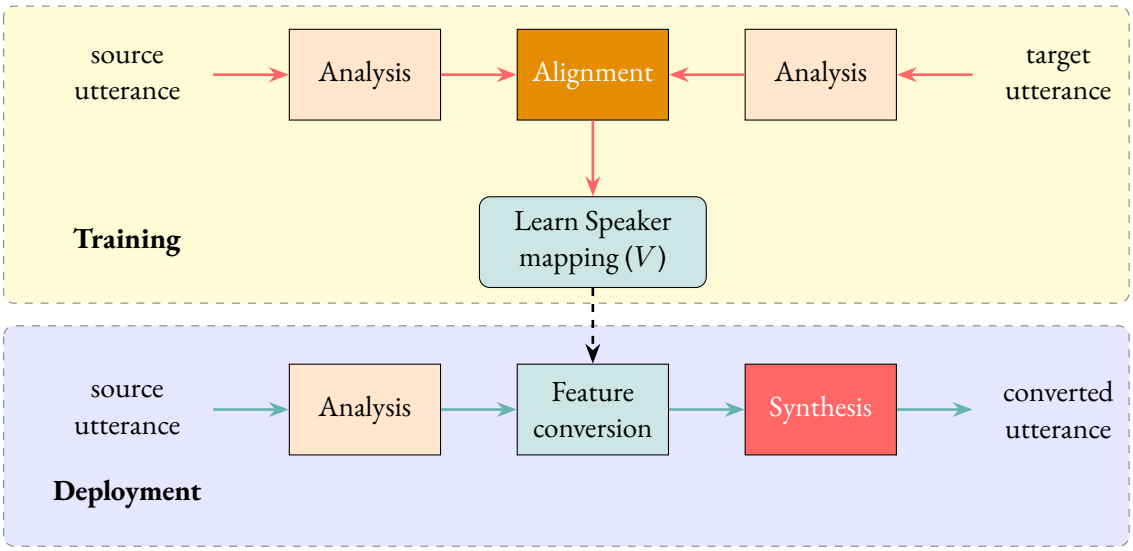


Fig. 2.16 General schema for traditional voice conversion with parallel data. Red arrows indicate training flow, while green arrows show the flow during conversion.

Figure 2.16 shows a general schema for traditional VC where red arrows show the training process of the speaker mapper. The orange blocks are parts of the vocoder (as described in Section 2.2.4) which provides algorithms to extract features, such as pitch, spectrogram, and aperiodicity from speech and re-synthesize the original signal, given these features. During the training phase, the parameters of the frame-wise mapping function V are learned to convert the source acoustic features into a given target speaker's acoustic features. To learn this mapping effectively, traditional VC approaches require a training set that contains parallel data, i.e., several different speakers uttering the same linguistic content. Parallel data reduces the complexity of VC mapping by keeping the same linguistic content in the source and target utterances, but it raises the issue of alignment because each speaker may have a different speaking rate. As an additional step during training, the source and target features are aligned in time using dynamic time warping (DTW) to deal with varying speaking rates. The learned function V is eventually used to transform the source features at test time to a particular target speaker's features frame-by-frame. A glaring limitation of this approach is that the parameters of V have to be learned separately for different source-target speaker pairs. And in order to scale this system, new speakers must be recorded while they utter the same sentences which already exist in the training data set.

Existing statistical models can be used to design the mapping function V, such as joint modeling of source and target using Gaussian mixture models [292] or exemplar dictionary-based source-target mapping [326, 90, 324]. Of course, deep learning has also been employed extensively to build the mapping function: feedforward networks [57], RNNs [209], and recently Transformer based models [125] have been proposed. The traditionally used DTW has also been replaced by the neural network attention mechanism [286] which takes the context into account and gives superior alignment between source and target features. However, the breakthrough in VC research has been achieved by *non-parallel* techniques that obviate the requirement for parallel data, such as CycleGAN-VC [145], which involves a generator $G_{S\to T}$ which maps source features to target features and a discriminator D_T that classifies whether the generated features belong to the target. $G_{S\to T}$ and D_T are jointly trained to learn an efficient mapping function. Since the source and target utterance may not contain the same linguistic content, another generator $G_{T\to S}$ and discriminator D_S are added to the architecture which learn the inverse mapping from target to source.

During the training phase, the source features are converted to the target and then back to the source in a cycle, and the converted features are verified at each step using their corresponding discriminator. All four models are jointly trained in an end-to-end fashion using the cycle-consistency loss function so that the linguistic content is preserved.

One of the limitations of CycleGAN-VC is that it only supports one pair of source-target speakers per model. This limitation is alleviated using a simple modification in the architecture, where the generator converts source features to target conditioned upon a speaker embedding provided at training and test time. This architecture supports many-to-many voice conversion and is referred to as StarGAN-VC [144]. Another class of non-parallel VC is based on the idea of *disentanglement*, which means that during training it learns to decompose speech utterances such that the speaker and content information are perfectly separated from each other. At test time, source speaker information can be replaced by the target while copying the source content, and speech is re-synthesized using the new features to perform VC. Variational autoencoders [119, 121] have been used to learn speaker-independent content features, which can generate voice converted speech based on a target speaker one-hot vector. Several extensions to this approach have been proposed, where speaker embeddings are also learned to have a continuous representation [120], and better disentanglement is achieved using speaker adversarial training [126, 41]. Another interesting approach to disentangle speaker identity from content is to leverage the fact that the speaker information stays constant throughout the utterance, hence it can be removed from the content embeddings by using instance normalization which averages out the global statistics [40].

Despite the progress made by non-parallel and disentanglement approaches, VC systems suffer from a lack of training data and require careful tuning of parameters. Recently researchers have observed the similarities between the neural architectures of TTS, ASR, and VC, which have enabled them to propose a parameter sharing mechanism either through joint training of these systems or re-using certain components which may be well optimized for their task due to the availability of large data sets. Zhang et al. [339] jointly train TTS and VC systems in an encoder-decoder architecture by having two encoders, one for source text and another one for source speech, which are merged into a single decoder that takes the speaker-independent intermediate representation and produces speech in the voice of the target speaker. Zhang et al. [340] propose to train a state-of-the-art TTS system and then transfer the decoder of the TTS to the VC system while supervising the encoder output of the VC system to be similar to the encoder output of the TTS system in order to maintain speaker independence. The similarities between the goals of TTS and VC, and the fact that they produce speaker-independent features have allowed several such techniques to be proposed recently [337, 124, 187]. Park et al. [323] propose Cotatron VC which leverages speaker-independent linguistic features coming out of a pre-trained TTS system and targets the speaker's identity to convert the given content into the target speaker's voice. This approach is similar to phonetic posteriorgram-based techniques which we describe below.

Speaker-independent linguistic features can be produced not only using TTS, but also ASR. When extracted from the output layer of an ASR system, they are referred to as *phonetic posteriorgrams* since they are optimized to classify phonetic information. Phonetic posteriorgram features are similar to the BN features ($\mathbf{B}$) described in Section 2.2.3, except that phonetic posteriorgrams are obtained from the output layer of the ASR network. ASR systems have matured quite a lot due to decades of research in this domain. They are also trained on large data sets which are readily available for some popular languages. This makes them desirable for the extraction of rich linguistic features that are also assumed to be speaker-independent. One of the earliest approaches to use phonetic posteriorgrams for VC was proposed by Sun et al. [277], who used them to generate target acoustic features, but a full VC model is required to be trained for different target speakers. Tian et al. [291] extend [277] to propose a model averaging and adaptation technique which reduces the data requirement for scaling the VC system to new target speakers. Phonetic posteriorgram-based

VC approaches are gaining popularity [290, 344, 338], and in fact, we also use them for the approaches proposed in this thesis.

2.4 Machine learning based anonymization methods

As described in Section 1.1, personal data such as name, address, gender, ethnicity, health conditions, biometric markers, etc. are sensitive because firstly, they may reveal the true identity of a person that is a direct attack on their privacy, and secondly some of these attributes may be embarrassing for them if published. The goal of anonymization is to transform the personal data of individuals such that it can no longer be linked with their true identity while preserving the utility of the data. In this section, we describe some of the general approaches towards anonymization that have been proposed in the machine learning community, not specifically for speech data.

The earliest ML-based anonymization methods were designed to be applied on large databases [296, 207, 59] that gather sensitive personally identifiable data from users, such as medical records, user surveys, travel history, software usage, browser fingerprints, political opinions, sexual preferences, etc. Before publishing such private information, the database curator must ensure that it is *de-identified* or *pseudonymized*, i.e., sensitive attributes (such as social security number, name, address, etc.) are replaced with a random placeholder which can be reversed using a lookup table, or fully *anonymized*, which implies irreversible removal of attributes. Although it may seem that the database is sanitized by the naive approach of removal of sensitive attributes and can be safely released in public, in fact, the database can often be successfully de-anonymized (i.e., the users can be re-identified) using a combination of attributes leading to unique identifiers and/or auxiliary knowledge from public sources as shown in the case of sanitized movie reviews [210], social networks [15], computer networks [49] and DNA sequences [50]. The common attributes that are present in both the sanitized database and the publicly available sources and are used to shortlist the potential identities of anonymous persons are called quasi-identifiers. It has been shown that 87% of the population of the USA can be uniquely identified using just three quasi-identifiers: zip code, gender, and date of birth [280]. This naive approach is clearly not sufficient to unlink users' data from their identities, hence several formal models of privacy, such as k-anonymity [250] and differential privacy [70], have been proposed.

k-Anonymity The intuition behind these methods is that the machine learning models do not need precise information about each subject. Instead, they aim to infer some aggregate statistics about the population in the database, so releasing the data in its original format is not required. In the re-identification scenarios concerning movie reviews, health records, etc., the attacker would generally find a unique set of quasi-identifiers for an individual, which is the motivation for imposing k-anonymity [250] over tabular databases. It is defined as the property of the anonymized database such that there are at least k individuals for every set of quasi-identifiers in it. This gives the user the ability to hide in a crowd and the attacker's chances are reduced to $\frac{1}{k}$ to attribute a particular data point to the correct user. It is implemented using two basic operations: *suppression*, that is to remove or replace quasi-identifier values with placeholders (e.g., the trailing digits of a zipcode can be replaced by the * symbol), and *generalization*, that is to club together values in a particular column using ranges (e.g., age > 30). The anonymity gets stronger as the value of k increases, but of course, it implies a trade-off with the utility of the released data set.

There have been efforts to learn efficient machine learning models using a data set that has been anonymized using k-anonymity, by searching for optimal hyperparameters [21], efficient clustering of rows [180], and full-domain generalization [170]. Moreover, there are several extensions proposed to address issues in k-anonymity, e.g., the lack of diversity in the anonymized data set is handled by l-diversity [189], and the vulnerability posed by the distinct distribution of sensitive attributes is solved using t-closeness [176].

Indeed, such modification of the database implies a significant reduction in utility, and yet there remain several vulnerabilities in k-anonymity due to the lack of randomization in the aggregation and the query mechanism. This is the motivation for the differential privacy (DP) [70] paradigm, which provides the strongest privacy guarantees at present and is deployed at major corporations [47] including Apple, Google, and Microsoft. It was also used to publish population data for the 2020 US census [2].

Differential Privacy In its simplest form, DP can be explained as a randomized response [316] when a binary question is asked to the user with yes/no as the response. In this scenario, a coin is tossed each time a response is to be recorded. If the coin says heads, then the true response is recorded, and if the coin says tails, then another coin is tossed. Based on the outcome of the second coin, the response may be recorded as yes if heads or no if tails. The randomness involved in this mechanism allows the users to have some plausible deniability towards their response and hence protects their privacy to some extent. As the number of participants increases, the aggregate result for the question becomes more accurate. At this point it is to be observed that, while DP can be used to anonymize databases, more generally it is a property of the algorithm which executes query functions over databases and generates noisy responses to preserve privacy. This mechanism ensures that the output distribution of the DP algorithm is statistically indistinguishable, no matter whether the data of a subject is present in the database or not. Hence, for example, the participants of a DP-enabled survey can rest assured that their data does not make a significant difference in the aggregate response of the survey, and can go on to safely participate in it.

DP [69] provides a rigorous probabilistic way to quantify the privacy leakage of an information release process. DP also comes with strong mathematical properties and a powerful algorithmic framework [70]. For these reasons, DP and its variants have become the gold standard notion of privacy in machine learning and many other scientific fields. Traditionally, DP is defined using the notion of "neighbouring" databases which differ on at most one record. Let $\mathcal{D}$ be the universal set of databases, and $d, d' \in \mathcal{D}$ be two databases with $|d|_1$ and $|d'|_1$ as their respective sizes, then the ℓ_1 distance between them is given by $|d - d'|_1$, which measures how many rows differ between d and d'. Now we can define the criteria of response queried from these databases which will satisfy the DP guarantee.

Definition 1 (Differential privacy) *Let $\mathcal{A}$ be a randomized algorithm which executes queries over any arbitrary database belonging to $\mathcal{D}$, and let $\epsilon > 0$. We say that $\mathcal{A}$ is ϵ-differentially private (ϵ-DP) if for any $d, d' \in \mathcal{D}$ such that $|d - d'|_1 = 1$ and any $S \subseteq \mathrm{range}(\mathcal{A})$:*

$$\Pr[\mathcal{A}(d) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(d') \in S],$$

where the probabilities are taken over the randomness of $\mathcal{A}$.

DP essentially requires that the probability of any output does not vary "too much" (as captured by ϵ) when changing the input. The smaller ϵ, the stronger the privacy guarantee, hence epsilon is called the privacy budget or privacy leakage.

DP possesses a number of desirable properties. First, any function of an ϵ-DP algorithm remains ϵ-DP (*robustness to post-processing*). Second, one can easily keep track of the privacy guarantees across multiple analyses (*composition*). In particular, given K algorithms that satisfy ϵ-DP, executing them on the same data and releasing their combined outputs is $K\epsilon$-differentially private.

In this thesis, we use a variant of traditional DP, called the local-DP model [62] or the fully distributed model which is more suitable when there is no trusted third party to collect raw data. In this model, it is assumed that the users' private data is not aggregated in a central database; instead, each user is the sole owner of their data and they can respond to questions in a differentially private manner. This nullifies the

possibility of any data theft or malicious use by the database curator, hence it is considered as a stronger privacy model. Local-DP can be defined over individual data points that may originate from features of raw data or intermediate representations of a neural network.

Definition 2 (Local differential privacy) *Let $\mathcal{A}$ be a randomized algorithm taking as input a data point in some space $\mathcal{X}$, and let $\epsilon > 0$. We say that $\mathcal{A}$ is ϵ-local differentially private (ϵ-LDP) if for any $x, x' \in \mathcal{X}$ and any $S \subseteq \text{range}(\mathcal{A})$:*

$$\Pr[\mathcal{A}(x) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(x') \in S],$$

where the probabilities are taken over the randomness of $\mathcal{A}$.

It is to be noted that local DP is equivalent to standard DP for databases of size 1, and hence, x and x' are always neighboring.

A standard way to design differentially private algorithms is based on output perturbation. A basic approach is to rely on the Laplace mechanism, which consists in adding Laplace noise calibrated to the ℓ_1-sensitivity of the (non-private) function one would like to compute on the data [69].

Definition 3 (Laplace mechanism) *Let $f : \mathcal{X} \to \mathbb{R}^d$ and let the ℓ_1-sensitivity of f be defined as*

$$\Delta_1(f) = \max_{x,x' \in \mathcal{X}} |f(x) - f(x')|_1.$$

Let $\eta = [\eta_1, \ldots, \eta_d] \in \mathbb{R}^d$ be a vector where each $\eta_i \sim Lap(\Delta_1(f)/\epsilon)$ is drawn from the centered Laplace distribution with scale $\Delta_1(f)/\epsilon$. The algorithm $\mathcal{A}(\cdot) = f(\cdot) + \eta$ is ϵ-local DP.

This approach can also be used to randomize data points directly (*input perturbation*), which corresponds to the case where $\mathcal{X} \subseteq \mathbb{R}^d$ and f is the identity function. However, adding noise to raw data often destroys utility. A better strategy is to perturb carefully designed feature representations of the data, as done in existing work on image [169] and text [24, 188, 86]. Our proposed approach to add DP noise in speech features is presented in Section 6.1.

Membership inference attacks [12] are a privacy threat related to trained classifier models where an attacker can deduce if a given data point is present in the training set of the classifier. It has been shown [248] that DP can provably bound the accuracy of such attacks and enhance the privacy of these models considerably. Although DP provides strong mathematical guarantees, the addition of large random noise with the assumption of worst-case adversaries significantly reduces the utility and increases the sample complexity of the model [87, 63].

2.5 The speaker anonymization task

This thesis derives its relevance from Article 25 of the GDPR which mandates the data controller to implement all possible technical and organisational measures to ensure the protection of personal data following the principles of privacy by design and by default. More precisely, speech data is protected by Article 9 of the GDPR which prohibits the processing of biometric personal data that reveals the gender, racial or ethnic origin, or health indicators of the data subject. This restriction has a direct impact on the potential scientific and commercial advancements, therefore it is relaxed by Recital 26 of the GDPR which freely allows the processing of *anonymous* data that can no longer be used to identify the original data subject. Article 4(5) of the GDPR mentions "pseudonymization" as a possible technical measure to achieve privacy by design, which is the task of processing personal data such that it can no longer be attributed to the data

subject without using additional information that must be secured separately to safeguard the identity of the data subject. Such processing may prove to be risky if the so-called additional information is accessed by a malicious entity, hence a stronger approach, namely anonymization, is explored in this thesis. Although the GDPR does not explicitly mention anonymization, ISO/IEC 29100:2011 [131] defines it as the process by which the personally identifiable information is *irreversably* altered such that the original data subject cannot be identified directly or indirectly, either by the data controller alone or in collaboration with any other party.

As mentioned before, the goal of this thesis is to completely unlink a speaker's identity from their speech utterances while maintaining the usefulness of the signal. In a generalized sense, this task is perceived as privacy-preserving data publishing [91] in the literature which envisages a similar vision of releasing useful data sets in the public domain for scientific and commercial progress without compromising individuals' privacy. Users are the default owners of their data, therefore publishers must anticipate potential hostile activity against them using their data and take proactive measures to mitigate any such possibility. Fung et al. [91] present a comprehensive survey of traditional privacy-preserving data publishing approaches that are applicable to relational databases. They describe the actors involved in data publishing (user, data publishers, and recipients), the types of attacks and the effectiveness of privacy models against each attack, anonymization methods and their fundamental operations, and finally metrics to assess the performance of privacy-preserving data publishing methods in terms of privacy and utility.

Due to legal and technological awareness in the society, privacy-preserving data publishing approaches specific to speech data have gained prominence in the past decade and several different methodologies have been proposed to achieve anonymization[11] to some extent. Nautsch et al. [214] attempted to describe the legal and technological advances with respect to privacy-preserving speech processing, but they only mention the techniques related to cryptographic methods and their evaluation. The methods of speaker anonymization that are most relevant to this thesis can be divided into five categories based on the type of technology they use: noise addition, speech transformation, voice conversion, speech synthesis, and adversarial learning.

Anonymization attempts using noise addition Ahmed et al. [6] present an end-to-end ASR method that injects DP noise at various levels in the pipeline to protect the user's identity and publish only the anonymized transcription instead of the speech signal. Publishing the text content of spoken data in a privacy-preserving manner is helpful to safeguard speakers' identity, but it limits the usage of speech. Hashimoto et al. [113] experiment with a different range of bandpass filtered noise to be added to speech signal to degrade speaker recognition performance in terms of increase in EER and preserve intelligibility. Although this method allows publication of speech data, it is difficult to manually calibrate the noise for unforeseen conditions, e.g., different languages, ambient noise, etc.

Anonymization attempts using speech transformation Speech transformation is considered the most convenient anonymization method since it does not require large data sets for training machine learning models; instead, it can be performed using careful signal processing based manipulations of speech parameters. Cohen-Hadria et al. [45] perform anonymization of voice recordings by using a low pass filter to remove formants and inverting the MFCCs to obfuscate speaker information while preserving the acoustic scene. Qian et al. [234] transform the spectrogram frequency scale of original speech by applying a composition of two nonlinear functions with random parameters. The resilience of this technique is dependent on the secrecy of these parameters. They also perform sensitive keyword substitution to completely sanitize the

[11] Legally speaking, the term "anonymization" refers to a method that fully achieves this goal. Following [294], we use it in a broader sense to refer to a method that aims to achieve this goal, even when it has failed to do so.

speech for publishing. Patino et al. [223] present the latest speech transformation method as a baseline for the first Voice Privacy Challenge [293] where the pole angles of the linear prediction (LP) spectral envelope are altered using McAdams coefficient (α_M) [197]. Specifically, the filter coefficients for each frame are derived using LP source-filter analysis, which are then used to extract the real- and the complex-valued pole positions. Thereafter, the angle of the complex-valued poles is raised to the power of a pre-determined α_M, causing the associated formant spectrum to expand or contract. The new complex-valued pole positions and the unchanged real-valued poles are then converted back to filter coefficients, and combined with the residual source information to resynthesize an anonymized time-domain speech frame. Gupta et al. [105] significantly improved this work by proposing the modification of both the pole angles as well as the pole radii of the LP spectrum. Although the parameter manipulations performed by speech transformation methods seem perceptually reasonable, they are easy to break using machine learning methods [273], hence they provide weak protection against privacy attacks.

Anonymization attempts using VC Voice conversion (VC) methods are the earliest and most obvious choice for speaker anonymization since their goal is to transform the voice of a given speaker into that of another speaker whose voice characteristics are known beforehand. Jin et al. [139] present the first known approach towards speaker anonymization by transforming any given source speaker to a single target voice present in the Festival speech synthesis system [28], and show that it performs well in terms of drop in the speaker identification accuracy. Bahmaninezhad et al. [17] convert a given source speaker into the average of all speakers of the same gender. Pobar and Ipšić [228] pre-train a set of speaker transformations and identify the speaker at test time to select one of the corresponding transformations. These methods are hardly applicable in practice since they require the source speaker to be present in the training set of the VC system and, in the context of anonymization, the amount of speech from the original speaker is often limited to one utterance. To relax this constraint, Magariños et al. [190] find the closest source speaker in the training set and apply one of the corresponding transformations. Yoo et al. [329] presents a many-to-many CycleGAN variational autoencoder-based VC method for speaker anonymization which takes a one-hot vector for the target speaker present in the training set. They experiment with several distributions of training speaker proportions as the target but yet do not allow external speaker identities to be used at run time.

Anonymization attempts using speech synthesis Techniques based on speech synthesis have also been proposed to relax the requirement of having source speakers in the training set of the anonymization system. For instance, Justin et al. [142] transcribe speech into a diphone sequence and re-synthesize it using a single target. These methods suffer from three limitations. First, they still result in a limited set of target speakers or speaker transformations, which prevents the original speaker from choosing an arbitrary unseen speaker as the target. Second, using a real speaker's voice as the target raises ethical concerns. Third, the conversion of speech to a sequence of discrete tokens as in [142] is error-prone and destroys all the paralinguistic and extralinguistic attributes. This motivates the objective of converting the original speaker's voice into an arbitrary, imaginary *pseudo-speaker*'s voice without relying on a transcription step. Speaker embeddings such as x-vectors [269] provide the continuous representation needed to define and generate such pseudo-speakers. Fang et al. [79] address this objective using a speaker-independent speech synthesis system. They select x-vectors within an external pool of speakers and average them to obtain a target *pseudo-speaker* x-vector. This x-vector, along with a representation of the original linguistic and intonation contents, is provided as input to a neural source-filter (NSF) based speech synthesizer [312] to produce anonymized speech. Han et al. [110] extend the framework presented in [79] to select a single target x-vector at random within a maximum distance from the original x-vector that satisfies a privacy metric based on differential privacy. Although these techniques manage to alleviate the three limitations mentioned before, they use weak evaluation criteria with

an assumption that the attacker does not know about the usage and the parameters of the anonymization algorithm.

Anonymization attempts using adversarial learning Recently, there has been some research towards speaker anonymization using adversarial training which implicitly models speaker information and removes it from the intermediate representations of a neural network that is optimized for some utility task, thereby learning privacy-preserving model parameters. One of the first approaches in this direction is investigated as a part of this thesis which is described in detail in Chapter 4. Espinoza-Cuadros et al. [75] present an autoencoder-based approach, which reconstructs the speaker representation by adversarially removing source speaker information from it, and use the new representation as the target pseudo-speaker. In a similar context, Champion et al. [36] examine the hypothesis that the linguistic features used for speech synthesis contain speaker information and use speaker adversarial training similar to [272] to mask the identities.

2.6 Summary of techniques

In this section, we succinctly recall the relevance of the abovementioned techniques for this thesis and mention how exactly they are reused or adapted in the following chapters.

We first delve into the details of speech generation using the vocal tract, which makes it clear that each person has a personal physiology that reveals cues to substantiate his/her identity from the speech signal. The specific configurations of articulators, that generate the phoneme sounds, also exhibit personally identifiable characteristics. Although in this thesis we do not perform any phoneme-specific analysis with respect to privacy, such phenomena are crucial to understand that the speaker's identity is highly entangled with the linguistic properties of sounds, and may possess identity markers in terms of nativeness or accent [53]. These properties and identity markers are recognized by examining the speech signal in the time domain or the frequency domain. The time-domain signal is a useful digital representation of sound and it exhibits several interesting properties which enable us to define either the local units of speech, i.e., phonemes, or global speaker-related characteristics like speaking rate [164]. It is noteworthy to mention that the time-domain signal also contains the effect of ambient noise, the microphone quality, and the subtle variations in the vocal tract even to produce the same linguistic content. Due to the dominant effect of speaker characteristics among these factors, it is certainly attainable to infer the speaker's identity yet it is difficult to disentangle these variations and capture parameters in the time-domain signal.

Evidently, it is much easier to analyze the speech signal after transforming it into the frequency domain. The spectrogram and the features derived from the frequency domain representation, such as MFCCs, are widely used to identify phonemes, speakers, emotions, etc. These features exhibit numerical patterns that correlate with the perceptual properties of human speech, for example, the fundamental frequency correlates with the pitch, the harmonic structure in the spectrum indicates voicing, and the position of the formant peaks characterize a phoneme. The goal of this thesis is to efficiently identify features corresponding to speaker information and manipulate them such that the speech signal is no longer attributable to the original speaker. We mentioned previously that the speaker information is present throughout the signal as the source characteristic, and shows distinctive cues in the spectral tilt, the nasalization patterns, and the speaking rate of an utterance. Although a rule-based approach, using a combination of acoustic cues and spectral features, can be devised to isolate the factors causing speaker distinction in a controlled setting, it is infeasible to model unknown variations caused by the effect of ambient noise, emotional states, conversational settings, and languages. Hence, strong statistical models, such as deep neural networks, are employed to project speaker or phoneme-related information to a high-dimensional hidden space where discrimination and identification are easier.

TDNNs are well suited for modeling the temporal dependencies in a speech signal, hence, they are used to design effective ASR (except for the end-to-end systems) and ASI models. ASR models are extensively used for two main purposes in this thesis. First, they are used as a key building block of the anonymization pipeline proposed in this thesis. And second, they are used to evaluate the private representations generated by the proposed anonymization techniques in terms of intelligibility. ASI models are used for two main purposes: 1) to model speaker information and generate useful representations, such as the x-vectors, and 2) to test the resilience of the anonymization techniques against re-identification attacks. Chapter 4 investigates whether private representations can be generated by an ASR network when it is trained in a domain-adversarial manner with an ASI adversary. In addition, ASI evaluation metrics are adapted for measuring the degree of privacy protection as explained in the next chapter.

Techniques that allow the manipulation of speaker-related properties of speech and the generation of a new speech signal, such as speech synthesis, speech transformation, and voice conversion, are crucial for the anonymization approaches proposed in this thesis. The remaining chapters, except Chapter 4, leverage these techniques to replace the original speaker's identity in the speech signal with a new identity by manipulating either the inputs or the parameters of these techniques. Since they generate an intelligible speech signal, they can be directly evaluated in terms of their utility for the task of anonymization. The speech data set processed by these techniques can also be published and easily evaluated in terms of speaker re-identification attacks. Finally, we employ differential privacy based anonymization techniques to remove residual speaker information from the speech signal and provide formal guarantees of privacy protection.

Chapter 3

Privacy Evaluation using Informed Attackers

> Facts do not cease to exist because they are ignored.
>
> *Aldous Huxley*

In this chapter, we state the actors involved in the speaker anonymization process, and we define the actions they can make via a threat model that connects them. This threat model is then used as a guide to design and evaluate privacy protection schemes. Specifically, we propose the concept of malicious entities, i.e., attackers who possess a certain degree of knowledge about the anonymization scheme, and use this concept to rigorously evaluate the level of privacy protection achieved by the considered schemes. Further, we provide a brief account of the performance metrics that are employed to evaluate these schemes in terms of privacy and utility, followed by a detailed comparison of three privacy metrics to assess their ability to express useful information and their vulnerabilities. Finally, we conduct a preliminary study using one speech transformation and two voice conversion-based methods to establish the role of the actors in the threat model and validate the claim that 'knowledgable' attackers measure the level of privacy protection reliably.

In Section 3.1, we present the threat model, the actors and their goals, and discuss the notion of attackers' knowledge. Section 3.2 and 3.3 describe the VC techniques and strategies behind the anonymization schemes considered in this chapter. The performance metrics used to evaluate the level of privacy protection and utility achieved are mentioned in Section 3.4. Section 3.5 describes the experimental setup including data sets, VC algorithm settings and attacker designs. Different attackers are compared in Section 3.6, while different privacy metrics are compared in Section 3.7. Finally, Section 3.8 summarizes the main findings of this chapter and paves the road for the following chapters.

The investigations in this chapter are conducted in collaboration with Dr. Mohamed Maouche. The attack model, the concept of attackers' knowledge, the implementation of anonymization schemes, and the experiments to compare the attackers were primarily contributed by the author of this thesis, while the definition of privacy metrics and the experiments to compare them were largely contributed by Dr. Maouche.

3.1 Attack model and the notion of attackers' knowledge

As explained in Section 1.1, speech data exhibits biometric characteristic of human beings which must be protected to safeguard the identity of individuals. According to the ISO/IEC International Standard 24745 on biometric information protection [130], publicly available biometric references must be *irreversible* and *unlinkable* for full privacy protection. Such protection must be resilient to re-identification attacks that may be strengthened using auxiliary information about the data set or the privacy protection scheme.

Throughout this thesis, we consider the following threat model. *Speakers* process their voice through an *anonymization* scheme. This anonymization step takes as input one or more *private speech* utterances along with some configuration parameters, and outputs a new speech signal or some kind of derived representation. The transformed utterances from one or more speakers form an *anonymized public speech* data set that is processed by a third-party *user* for, e.g., ASR training/decoding or any other downstream task. Given a public data set of anonymized speech (or speech representation in the form of feature vectors) contributed by several speakers, an attacker attempts to find which utterances in this data set are spoken by a given speaker. To do so, the attacker compares every utterance in the data set with a sample of speech from that speaker, that was either recorded or found in some other data source. In addition, the attacker can leverage some knowledge about the anonymization scheme as shown by the red arrow in Figure 3.1.

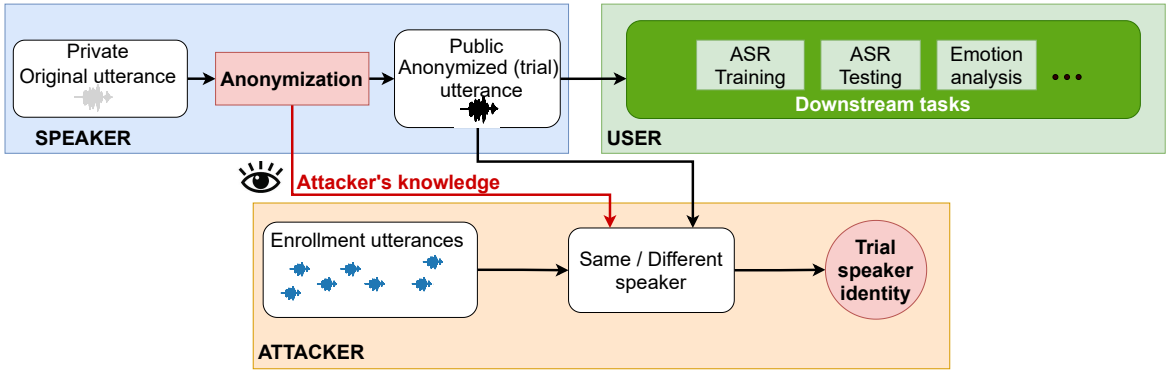


Fig. 3.1 Considered threat model. *Speakers* anonymize speech to conceal their identity before publication; *attackers* use biometric technology and knowledge of the anonymization scheme to re-identify them; *users* (e.g., speech technology companies) use the published data for downstream tasks such as ASR training.

Formally, an attacker has access to two sets of utterances: A (*enrollment/found data*) and B (*trial/public anonymized speech*), but knows the corresponding speakers in A only. The attacker designs a linkage function $LF(a, b)$ that outputs a score for any $a \in A$ and $b \in B$. Typically, this score is a similarity score obtained through a speaker verification system. The attacker then makes a decision whether a and b are *mated* (same speaker) or *non-mated* (different speakers) based on this score. A good speaker anonymization scheme must defeat such *linkage attacks* by concealing the speaker identity, while preserving the utility of speech for data *users* as measured for instance by the perceived speech naturalness and intelligibility and/or the performance of downstream tasks such as training an automatic speech recognition (ASR) system, thereby achieving a suitable privacy/utility trade-off. Figure 3.1 shows the three actors involved in this model, namely the *speaker*, the *attacker* and the *user*, along with their actions. The goals of the speaker and the user are intimately linked, while the attacker operates independently.

Crucially, all past studies have assumed a weak attack scenario where the attacker is unaware that anonymization has been applied to the public data [139, 142, 113, 234, 17, 79]. This raises the concern that the privacy protection may entirely rely on the secrecy of the design and implementation of the anonymiza-

tion scheme, a principle known as "security by obscurity" [202] that has long been rejected by the security community. There is therefore a strong need to evaluate the robustness of the anonymization to the knowledge that the adversary may have about the transformation. In practice, such knowledge may for instance be acquired by inspecting the code embedded in the speaker's device or in an open-source implementation.

As opposed to past studies, different linkage attacks are considered in this thesis depending on the attacker's knowledge of the anonymization scheme, as illustrated in Figure 3.2. At one end of the continuum, an *Ignorant* attacker is unaware of the speech transformation being applied, while at the other end an *Informed* attacker can leverage complete knowledge of the transformation algorithm and its parameters. In between, a *Lazy-Informed* attacker and a *Semi-Informed* attacker know the voice transformation algorithm but not its parameters and they exploit it to a different extent.

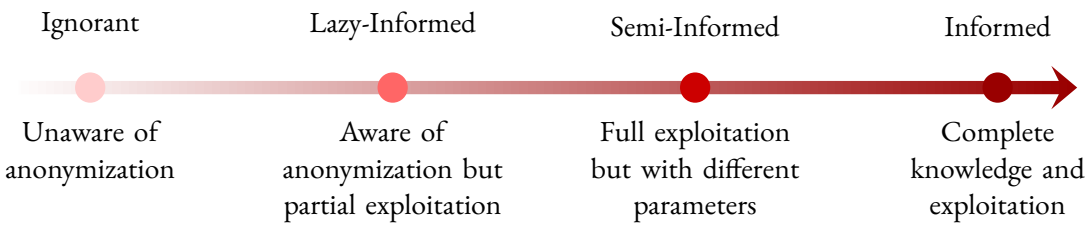


Fig. 3.2 Increasing degree of attacker's knowledge that determines the strength of the re-identification attack. Intermediate points on the continuum can be simulated as attack scenarios.

We have implemented several attackers depending on the choice of the VC algorithm and the target selection strategy as well as the extent of the attacker's knowledge (*Informed*, *Semi-Informed*, *Lazy-Informed* or *Ignorant*). This implementation is illustrated in Figure 3.3, where the first row indicates the baseline case 1 that uses the untransformed trial and enrollment sets along with the ASV_{eval} model trained on the original data.

2 Our *Ignorant* attacker is unaware of the VC step: he/she simply uses an i-vector+PLDA or x-vector+PLDA ASV_{eval} model trained on the untransformed training set and applies it to the untransformed enrollment set, while the trial set is anonymized.

3 Our *Lazy-Informed* attacker is aware of the anonymization scheme, i.e., the VC algorithm (see Section 3.2) and the target selection strategy (see Section 3.3.1), but not the particular choices of targets. He/she uses that knowledge to anonymize the enrollment set using a different choice of targets than the trial set. However, due to computational constraints, he/she does not retrain the ASV_{eval} model on anonymized data, and still considers an ASV_{eval} model trained on the untransformed training set for re-identification.

4 Our *Semi-Informed* attacker also knows the anonymization scheme but not the particular choices of targets. He/she advances one step ahead of the *Lazy-Informed* attacker and applies this strategy to the enrollment as well as the training sets by drawing random target speakers used by the VC method (we assume that the values of VC parameters are known to both *Lazy-Informed* and *Semi-Informed* attacker). As a result, the training and enrollment data are converted in a similar way as the trial data, but the target speaker associated with every speaker in the enrollment set is typically different from that which is associated with the same speaker in the converted trial set, except for the *const* speaker selection strategy (see Section 3.3.1) which uses a single target speaker for the entire data set. The training set is then used to train a new $\text{ASV}_{\text{eval}}^{\text{anon}}$ model to match the testing conditions during

the re-identification attack, i.e., both the x-vector/i-vector speaker representation extractor[1] and the PLDA speaker similarity score model are re-trained.

5 Finally, our *Informed* attacker has access to the actual VC models and target choices used to anonymize the trial set, so he/she converts the training and enrollment sets accordingly, and retrains the speaker verification model similar to the *Semi-Informed* attacker.

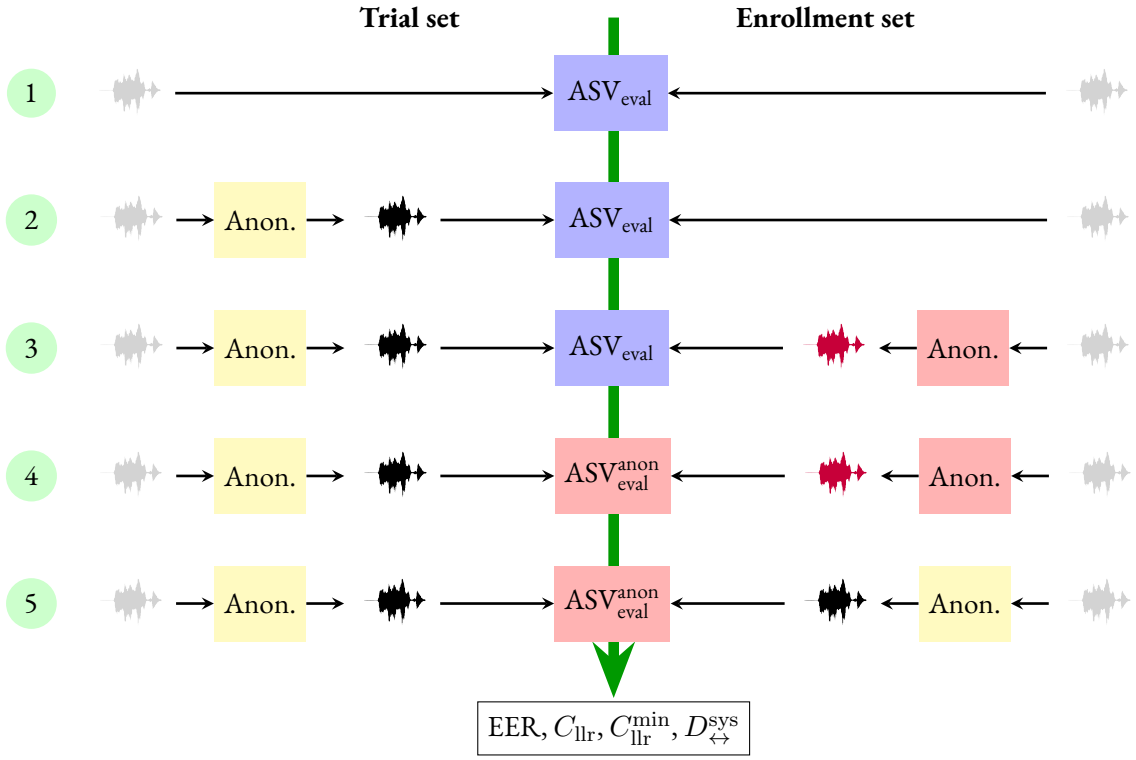


Fig. 3.3 Privacy evaluation in (1) Original, (2) *Ignorant*, (3) *Lazy-Informed*, (4) *Semi-Informed* and (5) *Informed* settings using ASV$_{\text{eval}}$ models. The blue ASV$_{\text{eval}}$ block indicates the model trained on original speech, the red ASV$^{\text{anon}}_{\text{eval}}$ block is trained on anonymized speech but with different parameters than the trial set. Although the ASV$^{\text{anon}}_{\text{eval}}$ model is always shown in red color, as per the explanation in Section 3.3.2, the *Semi-Informed* ASV$^{\text{anon}}_{\text{eval}}$ model differs from the model in the *Informed* setting when the *const* strategy is used.

In this chapter, we conduct preliminary investigations to assess the strength of *Ignorant*, *Semi-Informed*, and *Informed* attackers, and establish the validity of the proposed attack continuum. Chapter 5 additionally considers the *Lazy-Informed* attacker for evaluating the anonymization schemes. Our experiments evaluate three VC methods with different target speaker selection strategies in various attack scenarios to study unlinkability in the spirit of the ISO/IEC 30136 standard [132]. In each scenario, we measure how well each VC method protects the speaker identity against attackers that leverage state-of-the-art speaker verification techniques based on i-vectors [55] or x-vectors [269] to design linkage attacks. The *word error rate* (WER) achieved by a state-of-the-art end-to-end automatic speech recognizer [318] is also reported as measure of

[1]The extractor is a neural network in the case of x-vectors [269] and a joint factor analysis model in the case of i-vectors [55].

utility for the data user. While a formal listening test is beyond the scope of this thesis, a few samples of converted speech are available for informal comparison.[2]

3.2 Voice conversion methods

The criteria for selecting the VC methods for the experiments performed in this thesis are that they must be **1) non-parallel**, i.e., do not require a parallel corpus of sentences uttered by both the source and target speakers for training — this is important from both privacy and technical perspectives. The privacy risk arises due to the scarcity of parallel corpora which limits the data publisher to choose among a few openly available targets, thereby increasing the risk of an inversion attack. Furthermore, it requires the source speaker at test time to be present in the parallel corpus, which limits the usage of such a system to a selected few speakers; **2) many-to-many**, i.e., allow conversion between arbitrary sources and targets so that any speaker in a large corpus can be selected as the target, thereby increasing the choices for anonymization targets ; **3) source- and language-independent**, i.e., do not require enrollment sentences for the source speaker and do not rely on language-specific ASR or phoneme classification — this is important from a usability perspective as it frees the speaker from the burden of enrolling and it is applicable to any language (including under-resourced ones), and from a privacy perspective since enrollment translates into the storage of a speaker's biometric identity which poses even greater privacy threats.

The first criterion is a general requirement which all speech anonymization methods must satisfy. This criterion and the second one are satisfied by the methods presented in the following chapters too. The third criterion is quite strict: many VC methods, such as StarGAN-VC [144] or the ASR-based method in [79], do not satisfy it. We found that the vocal tract length normalization (VTLN) based methods in [233, 279] and the one-shot method in [40] satisfy all criteria. In this chapter, we use models trained over English speech [220] but do not use any other linguistic resources such as transcriptions, hence in principle they may be applicable to other languages as well. A general description of the three selected VC methods is provided here and the details of their implementation are presented later in Section 3.5.2.

3.2.1 VoiceMask

VoiceMask is described in [233] as the frequency warping method based on the composition of a log-bilinear function, expressed as

$$f(\omega, \alpha) = \left| -i \log \frac{\mathrm{e}^{i\omega} - \alpha}{1 - \alpha \mathrm{e}^{i\omega}} \right|, \tag{3.1}$$

and a quadratic function, given by

$$g(\omega, \beta) = \omega + \beta \left(\frac{\omega}{\pi} - \left(\frac{\omega}{\pi} \right)^2 \right). \tag{3.2}$$

Here $\omega \in [0, \pi]$ is the normalized frequency, $\alpha \in [-1, 1]$ is the warping factor for the bilinear function, and $\beta > 0$ is the warping factor for the quadratic function. Therefore, the warping function is of the form $g(f(\omega, \alpha), \beta)$. The two parameters, α and β, are chosen uniformly at random from a predefined range which is found to produce intelligible speech while perceptually concealing the speaker identity.[3] In the following, we apply this transform to the spectral envelope (see Section 2.2.1) rather than the pitch-

[2] https://github.com/brijmohan/adaptive_voice_conversion/tree/master/samples

[3] Figure 9 in [233] shows the effect of α and β on the voice distortion.

synchronous spectrum as in the original paper.[4] In addition, we apply logarithm Gaussian normalized pitch transformation [182] so as to match the pitch statistics of a target speaker. Let $\mathbf{p}_{\text{src}}$ represent the pitch sequence of the source speaker, μ_{src} and σ_{src} be the mean and standard deviation of the source speaker, and μ_{tgt} and σ_{tgt} be the mean and standard deviation of the target speaker. Then the transformed pitch sequence can be obtained as

$$\log(\mathbf{p}_{\text{tgt}}) = \frac{\log(\mathbf{p}_{\text{src}}) - \mu_{\text{src}}}{\sigma_{\text{src}}} \times \sigma_{\text{tgt}} + \mu_{\text{tgt}}. \tag{3.3}$$

The authors claim that this transformation is difficult to inverse when the parameter values are unknown because they are randomly selected from a large interval. However, VoiceMask uses the same parameter values to warp the spectra at each time step of the utterance. Therefore, this approach is quite limited to conceal the identity of the source speaker and to mimic the target speaker.

3.2.2 VTLN-based voice conversion

VTLN-based VC [279] represents each speaker by a set of centroid spectra $\{\overline{\mathcal{F}}_1, \ldots, \overline{\mathcal{F}}_K\}$ extracted using the CheapTrick [205] algorithm for K pseudo-phonetic classes. These classes are learned in an unsupervised fashion by clustering all speech frames of all utterances from this speaker. For each class of the source speaker k, the procedure finds the class of the target speaker k' and the warping coefficient ϑ that minimize the distance between the source centroid spectrum transformed using a spectral conversion function $F_\vartheta(\overline{\mathcal{F}}_{\text{s},k}, \omega) = \overline{\mathcal{F}}_{\text{s},k}(\tilde{\omega}_\vartheta(\omega))$ and the target centroid spectrum $\overline{\mathcal{F}}_{\text{t},k'}$:

$$\vartheta, k' = \underset{\vartheta', k''}{\operatorname{argmin}} \int_{\omega=0}^{\pi} \left| \overline{\mathcal{F}}_{\text{t},k''} - F_{\vartheta'}(\overline{\mathcal{F}}_{\text{s},k}, \omega) \right|^2 d\omega. \tag{3.4}$$

All speech frames in that class are then warped using a power function [73]

$$\tilde{\omega}_\vartheta(\omega) = \left(\frac{\omega}{\pi}\right)^\vartheta. \tag{3.5}$$

Similarly to above, we apply this warping to the spectral envelope and also perform Gaussian normalized pitch transformation (Eq. (3.3)) so as to match the pitch statistics of the target. Compared to VoiceMask, this approach warps the frequency axis in different directions over time. The parameters of this method include the number of classes K and the chosen target speaker.

Although this algorithm does not require parallel data, it necessitates the storage of a set of K centroid spectra for each source or target speaker. Increasing the value of K might enhance the quality of the output speech because distinct phonetic classes may obtain their own set of warping coefficients ϑ rather than averaged coefficients over similar classes. It might also have some effect on the strength of anonymization due to an increase in the number of values of ϑ within an utterance, i.e., the frequency axis can be warped in more directions than before. Within the scope of this chapter, we fix the value of K based on the author's recommendations [279] and do not investigate the effect of changing it.

3.2.3 Disentangled representation based voice conversion

The third approach is based on disentangled representation of speech as proposed in [40, 301]. The core idea is that speaker information is statically present throughout the utterance but content information is dynamic. This approach is based on a neural network transformation and uses a *speaker encoder* and a *content*

[4] Strictly speaking, VoiceMask is a voice transformation method rather than a VC method: pitch is converted from the source speaker to a target speaker, but the spectral envelope is not related to a particular target speaker.

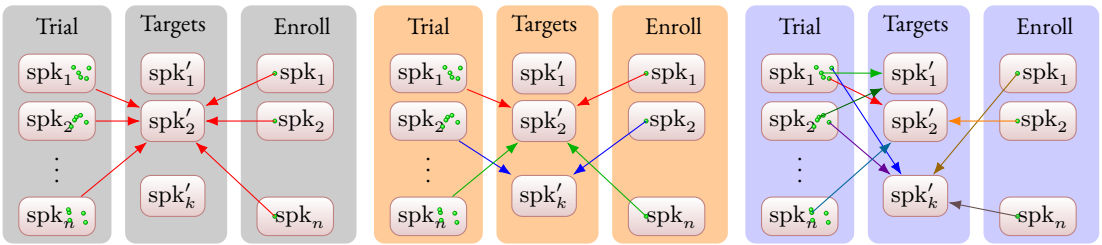
encoder to separate the factors of variation corresponding to speaker and content information. The only parameter of this method is the chosen target speaker.

3.3 Target selection strategies and exploitable parameters

In the following, we consider that the VC function and the sets of possible parameter values are known to all actors. Three parameter selection (a.k.a. target selection) strategies are defined for the three VC methods above, which can be seen as key ingredients of a "private-by-design" speech processing system. Thereafter the knowledge that an attacker trying to compromise the system could have about the VC function and the target selection strategy is described.

3.3.1 Target selection strategies

Three possible target selection strategies are considered which act as the core part of the defense against re-identification attacks, hence they are also called privacy protection strategies. They are introduced in order of increasing ramdomness. In the *const* strategy as shown in Figure 3.4(a), the VC function is constant across all speakers and all utterances. This means choosing a unique target speaker and, in the case of VoiceMask, fixed values for α and β. In the *perm* strategy depicted in Figure 3.4(b), the conversion parameters are chosen at random once by each speaker. In other words, when a speaker downloads the VC module on his/her device, he/she selects a personal target speaker and, in the case of VoiceMask, personal random values for α and β. Finally, in the *random* strategy illustrated in Figure 3.4(c), each time a speaker applies VC to an utterance, a random set of parameters is drawn, i.e., a random target speaker is selected and, in the case of VoiceMask, random values are drawn for α and β.



(a) *const* (b) *perm* (c) *random*

Fig. 3.4 Three target selection strategies: *const*, *perm* and *random*. Trial utterances are taken from the public, anonymized data set, while enrollment utterances are found data used by the attacker. Utterances are shown by small green balls, and the arrows indicate the mapping between original and target speakers.

3.3.2 Exploitable parameters

As explained in Section 3.1, the effectiveness of anonymization is evaluated using three types of attackers based on the extent of their knowledge about the VC function and its exploitable parameters. An *Ignorant* attacker is not aware that VC has been applied at all. In contrast, an *Informed* attacker knows the VC method and its exact parameter values (i.e., the chosen target speaker and the values of α and β). One may argue that an *Informed* attacker is not very realistic (except for the *const* strategy), while an *Ignorant* attacker is very weak. Between these two extreme cases, the *Semi-Informed* attacker knows the chosen VC method (VoiceMask, VTLN, or disentangled representation) and the target selection strategy (*const*, *perm*, or *random*), but not the actual target (i.e., the actual target speaker or the value of α and β). This is arguably more realistic since

the VC algorithm and the target selection strategy may be open-source, while (except for the *const* strategy) the target chosen by the speaker is much less easily accessible.

For the *perm* and *random* strategies, the advantage of an *Informed* attacker over a *Semi-Informed* attacker only comes from anonymizing the enrollment data using the same targets as the trial set. Indeed, the *Informed* attacker cannot anonymize the training set using the same (source,target) pairs as the enrollment and trial sets because the source speakers are disjoint. Therefore, the two attackers essentially use the same $\text{ASV}_{\text{eval}}^{\text{anon}}$ model. In the *const* case, the advantage comes not only from anonymizing the enrollment data using the same (unique) target as the trial set, but also from re-training the model over the training set that has been anonymized using the same target as well. Hence, the $\text{ASV}_{\text{eval}}^{\text{anon}}$ model for *const +Informed* is distinct from *const +Semi-Informed*, but for other strategies they do not differ (except with different random choices of target speakers).

3.4 Performance metrics

Historically, the usual metrics employed in the speaker verification community have been used to assess the (in)ability of an attacker to recognize the speaker, which is considered as a proxy for the degree of privacy protection. On the other hand, utility is considered a nebulous concept because it depends on the user, who may want to use the anonymized speech for transcription, scientific analysis, or broadcast purposes. Here, concrete privacy and utility metrics are presented which are used throughout this thesis to evaluate the proposed speaker anonymization schemes.

3.4.1 Privacy measures

In the context of the attacker's continuum, privacy may be measured as the deterioration in the attacker's ability to identify the original speaker from the new representation. If the original speaker is present in the training set of the system, then ASI, i.e. *closed-set* identification, can be used over anonymized speech to quantify the decrease in accuracy, precision, and recall for the speaker. If the speaker is not present in the training set, then ASV, i.e. *open-set* authentication, is used instead and the gain in privacy is proportional to the increase in the confusion of the system to identify the original speaker. The open-set evaluation is much closer to the realistic scenario where an attacker might obtain a small amount of speech data and use it to enroll the speaker under attack. Therefore, ASV systems are often used in this thesis to implement concrete attacks.

The most widely used ASV metric is the *equal error rate* (EER): it considers an attacker that makes a decision by comparing speaker similarity scores with a threshold and it assigns the same cost to false alarms and misses [133]. The *application-independent log-likelihood-ratio cost function* $C_{\text{llr}}^{\text{min}}$ generalizes the EER by considering optimal thresholds over all possible priors and all possible error costs [32]. In the following, we consider a third metric called *linkability* which has recently emerged from the biometric template protection community but has received little attention in the speech community so far [98]. This metric, denoted as $D_{\leftrightarrow}^{\text{sys}}$, estimates the distributions of scores for mated vs. non-mated trials and computes their overlap. These metrics are formally defined below.

Equal Error Rate (EER) The EER is the classical metric used in speaker recognition. It assumes a threshold-based decision on the score. If $LF(a, b)$ is greater than a certain threshold τ, the two utterances a and b are considered to be mated. Two types of errors can be made: false alarms with rate $P_{\text{fa}}(\tau)$, and misses with rate $P_{\text{miss}}(\tau)$. The EER is the error rate corresponding to the threshold τ^* for which the two types of errors are equally likely:

$$\text{EER} = P_{\text{miss}}(\tau^*) = P_{\text{fa}}(\tau^*). \tag{3.6}$$

Log-Likelihood-Ratio Cost Function C_{llr} and $C_{\text{llr}}^{\text{min}}$ C_{llr} is also a common speaker recognition metric [32]. It is *application-independent* in the sense that it pools across all possible costs for false alarm vs. miss errors, and all possible priors for mated vs. non-mated trials. Let M (resp., $\overline{M}$) be the set of mated (resp., non-mated) trials and $|M|$ (resp., $|\overline{M}|$) its cardinality. Denoting by $\text{llr}(p)$ the log-likelihood ratio of mated vs. non-mated hypotheses for trial $p = (a, b)$, C_{llr} is defined as

$$C_{\text{llr}} = \frac{1}{\log 2}\left[\frac{1}{|M|}\sum_{p\in M}\log\left(1+e^{-\text{llr}(p)}\right)+\frac{1}{|\overline{M}|}\sum_{p\in\overline{M}}\log\left(1+e^{\text{llr}(p)}\right)\right]. \tag{3.7}$$

C_{llr} assesses the overall detection which includes both discrimination, i.e., the distinction between speakers, and calibration, i.e., the selection of an optimal decision threshold. In practice, discrimination alone is more relevant as a privacy metric. To measure it, a derived metric called $C_{\text{llr}}^{\text{min}}$ can be computed by optimal calibration of the scores $LF(p)$ into log-likelihood ratios using a monotonic increasing transformation. This transformation is found via the Pool Adjacent Violators algorithm (PAV), see [304] for details.

Linkability A linkability metric was proposed in [98] for biometric template protection systems. This metric can be generalized for any two sets of items. Denoting by H (resp., $\overline{H}$) the binary variable expressing whether two random utterances a and b are mated (resp., non-mated), the local linkability metric for a score $s = LF(a, b)$ is defined as $p(H \mid s) - p(\overline{H} \mid s)$. When the local metric is negative, an attacker can deduce with some confidence that the two utterances are from different speakers. The authors of [98] argued that the local metric should estimate the strength of the link described by a score rather than measure how much a score describes non-mated relationships. Therefore they proposed a clipped version of the difference:

$$D_{\leftrightarrow}(s) = \max(0, p(H \mid s) - p(\overline{H} \mid s)). \tag{3.8}$$

The global linkability metric $D_{\leftrightarrow}^{\text{sys}}$ is the mean value of $D_{\leftrightarrow}(s)$ over all mated scores:

$$D_{\leftrightarrow}^{\text{sys}} = \int p(s \mid H) \cdot D_{\leftrightarrow}(s)\, ds.$$

In practice, $D_{\leftrightarrow}(s)$ is rewritten as $(2 \cdot \omega \cdot \text{lr}(s))/(1 + \omega \cdot \text{lr}(s)) - 1$ where the likelihood ratio $\text{lr}(s)$ is $p(s \mid H)/p(s \mid \overline{H})$ and the prior probability ratio ω is $p(H)/p(\overline{H})$, and $p(s \mid H)$ and $p(s \mid \overline{H})$ are computed via one-dimensional histograms.

3.4.2 Utility measures

Generally, it is assumed that reasonably intelligible, natural-sounding, good quality audio is sufficient for any kind of utility. Humans can be employed to listen to the transformed speech data and rate these attributes on a perceptual scale, but this setup is costly and time-consuming. Instead, most speaker anonymization studies including ours use an ASR system as the objective judge of these attributes. It is a good proxy for measuring the utility since the given attributes highly correlate with ASR performance. ASR performance is measured in terms of the WER. The WER is computed by first aligning the estimated word sequence with the reference word sequence, and then counting the number of substitutions (S), deletions (D), and insertions (I) required to convert the estimates to the references, where each reference utterance i is of length $L^{(W_i)}$. The WER for the whole data set is given by the following formula:

$$\text{WER} = \frac{S + D + I}{\sum_i L^{(W_i)}}. \tag{3.9}$$

3.4.3 Comparison of privacy metrics

In the later part of this chapter, experiments are performed to assess the suitability of the three metrics, EER, $C_{\text{llr}}^{\text{min}}$ and $D_{\leftrightarrow}^{\text{sys}}$, for the evaluation of speaker anonymization. In addition to comparing the metrics in their form and substance, simulated data is generated to exhibit their blindspots. Experiments are also conducted on real speech data processed by the anonymization schemes considered in Section 3.3.1 against the three different attackers defined in Section 3.3.2. Overall, the aim is to understand the complementary factors underlying different metrics and ensure that the anonymization schemes being evaluated were not designed to fool attackers that follow one specific speaker verification method but would fail with others.

Based on the definitions in Section 3.4.1, it is evident that the three metrics do not provide the same information. Both the EER and $C_{\text{llr}}^{\text{min}}$ measure the probability of error of an attacker that makes decisions based on a threshold on the linkage function (one particular threshold for EER and all possible ones for $C_{\text{llr}}^{\text{min}}$). Linkability measures something different: it evaluates how different the distributions of mated vs. non-mated scores are. There is no attacker making a decision and there is no threshold or, from another perspective, the best possible *oracle* attacker (not necessarily threshold-based) is assumed. In addition, if we consider how general are the metrics, on the one hand $C_{\text{llr}}^{\text{min}}$ is a direct extension of the EER as it does not focus on one single threshold. On the other hand, $D_{\leftrightarrow}^{\text{sys}}$ is evaluated over all the encountered mated scores. In Section 3.7, we provide experimental examples that highlight the impact of these differences.

3.5 Experimental setup

In this section, we describe in detail the data sets, the parameters of the VC methods, and the models used to evaluate privacy and utility.

3.5.1 Data and evaluation setup

The majority of experiments in this thesis are performed on the openly available LibriSpeech corpus [220] which is a 960-hour data set containing read English speech derived from a large collection of audiobooks in the public domain.[5] The audiobooks are recorded by volunteers in rather clean ambient conditions using their own microphone device, hence the recording conditions may differ from speaker to speaker. The audio is sampled at 16 kHz and sufficient linguistic resources, such as the lexicon and the language models, are made available for download which makes it suitable for training ASR models. The corpus is gender-balanced in terms of the number of speakers and their individual duration as shown in Table 3.1, therefore it can be well suited for training speaker identification models. The whole data set is divided into evaluation and training subsets with further segregation of utterances incurring lower WER, designated as "clean", from the ones incurring higher WER, i.e., "other". Careful selection has been done to maintain the duration of individual utterances to about 10 s and to balance the data such that the per-speaker duration within the subset is almost the same.

For the experiments in this chapter, the 460-hour clean training set (*train-clean-100* + *train-clean-360*), which contains 1,172 speakers, is used to train the disentanglement transform. Out of the *test-clean* set, an *enrollment* set (438 utterances) and a *trial* set (1,496 utterances) are created with different utterances from the same 29 speakers (13 male and 16 female, not in the training set) considered as source speakers. The details of the trial set are shown in Table 3.2. The target speakers for all three VC methods are randomly picked from the training and *test-clean* sets.

For each VC method and target selection strategy, all utterances in the trial set are mapped to possibly different target speakers in the training or trial set. The converted trial set serves as the public anonymized

[5] https://librivox.org/

Table 3.1 Subsets of the LibriSpeech data set along with their total duration in hours, duration per speaker in minutes, and number of male and female speakers.

	Subset	Duration (h)	per speaker (min)	Male speakers	Female speakers
Evaluation	dev-clean	5.4	8	20	20
	test-clean	5.4	8	20	20
	dev-other	5.3	10	17	16
	test-other	5.1	10	16	17
Training	train-clean-100	100.6	25	126	125
	train-clean-360	363.6	25	482	439
	train-other-500	496.7	30	602	564

Table 3.2 Detailed description of the trial set for speaker verification experiments.

	Male	Female
Number of Speakers	13	16
Number of Mated trials	449	548
Number of Non-mated trials	9,457	11,196

data set that attackers want to de-anonymize by designing a linkage attack. To this end, attackers have access to the enrollment set which serves as the found data used to model the speakers in the trial set.

The attackers also have access to the 460-hour training set to train state-of-the-art speaker verification methods based on x-vectors [269] and i-vectors [55], which are stronger than the Gaussian mixture model-universal background model (GMM-UBM) based method used in the seminal work of [139]. We adapt the *sre16* Kaldi recipe for training x-vectors and i-vectors to LibriSpeech.[6] The recipe is customized to use a smaller network architecture for x-vector computation than the original architecture presented in Table 2.1. Specifically, compared to that architecture, the *frame4*, *frame5* and *segment7* layers are removed, thereby also reducing the *stats pooling* layer to $512T \times 1024$ and the *segment6* layer to 1024×512. Here T refers to the utterance-level context. This reduced architecture performs slightly better on LibriSpeech than the original one. We use the PLDA score given by Equation (2.26) to compute the similarity between speakers. The PLDA model is again estimated over the x-vectors extracted from the 460-hour training set.

Finally, the utility of each VC method is evaluated in terms of the resulting $\text{ASR}_{\text{eval}}^{\text{anon}}$ performance, that is trained and tested on the converted data (Figure 3.5, case 3), and this WER is compared with the baseline WER obtained on the clean speech using ASR_{eval} (Figure 3.5, case 1). A hybrid connectionist temporal classification (CTC) and attention based encoder-decoder [318] is used to build ASR_{eval}, that is trained on the converted 460-hour training set using the standard recipe for LibriSpeech provided in ESPnet[7]. This model is also used for the experiments in the next chapter.

3.5.2 Voice conversion settings

VoiceMask. Pitch, aperiodicity and spectral envelope are extracted using the pyworld vocoder[8]. Only the *random* strategy is followed: other target selection strategies have not been applied because fixed values for α and β (whether speaker-dependent or not) are prone to inversion attacks. The value of α is uniformly

[6] https://github.com/brijmohan/kaldi/tree/master/egs/librispeech_spkv/v2

[7] https://espnet.github.io/espnet/

[8] https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder

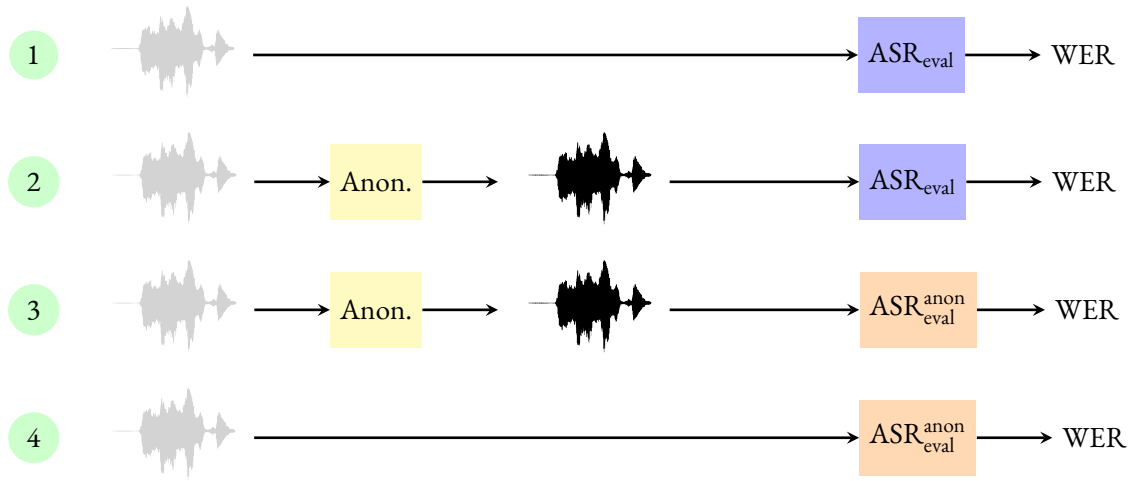


Fig. 3.5 Utility evaluation of 1 original speech data (OO) and 2 anonymized speech data (AO) using the ASR_{eval} model that is trained over original speech. Case 3 indicates that the $\text{ASR}_{\text{eval}}^{\text{anon}}$ model used for decoding anonymized speech is re-trained over anonymized speech data (AA). Similarly, in case 4 the $\text{ASR}_{\text{eval}}^{\text{anon}}$ model is used to decode original speech (OA). The yellow block indicates the application of the anonymization scheme.

sampled such that $|\alpha| \in [0.08, 0.10]$ then β in $[-2, 2]$ such that $0.32 \leq \text{dist}_{f_{\alpha,\beta}} \leq 0.40$ where $\text{dist}_{f_{\alpha,\beta}} = \int_0^{\pi} |f_{\alpha,\beta}(\omega) - \omega| d\omega$ is the distortion strength of the warping function. Although our implementation slightly differs from the original method as described in Section 3.2, we use the ranges for α and β provided by VoiceMask's authors in [233] since they produce most intelligible output. A subset of 100 target speakers is randomly selected and, for every utterance, pitch is transformed so as to match a random speaker within that subset.

VTLN-based VC. Pitch, aperiodicity and spectral envelope are extracted using the pyworld vocoder. For each speaker, speech frames are selected using energy-based voice activity detection (VAD) with a threshold of 0.06, and their spectral envelope are clustered via k-means with $K = 8$. In strategy *const*, only one target speaker is selected. In *perm*, a random subset of 100 target speakers is drawn and, for each source speaker, a random target is selected within the subset. In *random*, a random subset of 100 target speakers is drawn and, for each source utterance, a random target within the subset is selected.

Disentangled representation based VC. A publicly available implementation of this method is used.[9] As per the authors' suggestion in the preprocessing script, the disentanglement model (which includes a speaker encoder, a content encoder, and a decoder) is trained over the *train-clean-100* subset of the LibriTTS corpus (itself a subset of the 460-hour training set of LibriSpeech), with a batch size of 128 and a learning rate of 0.0005 for 500,000 iterations. All three target selection strategies are applied similarly to VTLN-based VC except that only the source utterance and one random utterance from the target speaker are used as inputs to the content and speaker encoders, respectively. Other utterances from the source and target speakers are unused.

[9] https://github.com/jjery2243542/adaptive_voice_conversion

3.6 Experimental comparison with different attackers

The three different VC-based anonymization schemes described in Section 3.3.1 are evaluated against the three attackers defined in Section 3.1. First and foremost, the ASR_{eval} and ASV_{eval} systems are trained and applied to the original (untransformed) data for baseline performance. EERs are obtained over the trial set described in Table 3.2 for i-vector and x-vector based verification, and WERs for ASR_{eval} and $\text{ASR}_{\text{eval}}^{\text{anon}}$ over the evaluation subsets of LibriSpeech mentioned in Table 3.1.

Tables 3.3 and 3.4 present the EER for x-vector and i-vector based speaker verification for the three attackers and the various VC methods and target selection strategies. Interestingly, the *Informed* attacker achieves similar or even slightly lower EER than the baseline in most cases. This indicates that, when the attacker has complete knowledge of the VC scheme and target speaker mapping, none of the VC methods can protect the speaker identity. While an attacker with such complete knowledge is not very realistic in most practical cases, our results show that speaker information has not been totally removed and is somehow still present in the converted speech. They also indicate that privacy protection only relies on the randomization introduced by the target selection strategies.

Table 3.3 EER (%) achieved using x-vector-PLDA based speaker verification.

	VoiceMask	**VTLN-based VC**			**Disentangl.-based VC**		
Attackers ↓ / Strategies →	*random*	*const*	*perm*	*random*	*const*	*perm*	*random*
Informed	5.01	4.71	3.91	6.32	4.71	0.20	5.52
Semi-Informed		12.84	23.37	6.32	13.64	43.03	5.42
Ignorant	28.69	24.27	30.99	27.38	27.68	32.20	30.59
Baseline	4.31						

Table 3.4 EER (%) achieved using i-vector-PLDA based speaker verification.

	VoiceMask	**VTLN-based VC**			**Disentangl.-based VC**		
Attackers ↓ / Strategies →	*random*	*const*	*perm*	*random*	*const*	*perm*	*random*
Informed	8.22	6.22	10.23	9.84	4.71	0.20	11.03
Semi-Informed		18.25	31.49	18.76	15.65	43.93	10.53
Ignorant	50.55	26.08	49.15	49.15	49.95	47.74	49.85
Baseline	4.61						

For the more realistic *Semi-Informed* attacker, it is observed that strategy *perm* is quite effective in protecting privacy and shows the highest gains in EER. This is because the (source, target) speaker pairs in the enrollment set may not be the same as in the trial set, hence greater confusion is induced during inference. It is also important to note that strategy *random* is not much affected by the change of speaker mapping, which is intuitive because in this case the utterances are already being mapped randomly to different speakers. In such a case, *Semi-Informed* attacker is expected to perform as good as the *Informed* attacker as seen in the results above, with an exception of i-vectors trained over utterances anonymized using VTLN-based VC scheme. While further investigation is required to understand this surprising result concerning i-vectors, such investigation is beyond the focus of this thesis, which concentrates on x-vectors in the following due to

their superior re-identification performance. Strategy *const* is also slightly affected by the change of mapping because the target speakers used by the attacker for anonymizing the training and enrollment sets are not the same as the target speakers used to anonymize the trial set, but the effect is not as significant as strategy *perm*. A preliminary *Lazy-Informed* experiment was also performed with the VoiceMask technique and evaluated using the baseline x-vector ASV_{eval} model. The obtained EER of 20.96% lies in-between the *Informed* (5.01%) and the *Ignorant* (28.69%) attackers. This scenario is explored in more detail in Chapter 5.

Consistently with past results in the literature, the *Ignorant* attacker performs worst in terms of EER. This confirms that, when the attacker is oblivious to the privacy-preserving mechanism, we can protect speaker identity completely. Figure 3.6 shows the distribution of i-vector PLDA scores for mated and non-mated trials, i.e., the uncalibrated log-likelihood ratios between *same-speaker* and *different-speaker* hypotheses. For full unlinkability, the distributions of mated and non-mated scores must be identical. We observe that the overlap between the two distributions decreases as we move from the *Ignorant* to the *Informed* attacker, hence increasing linkability.

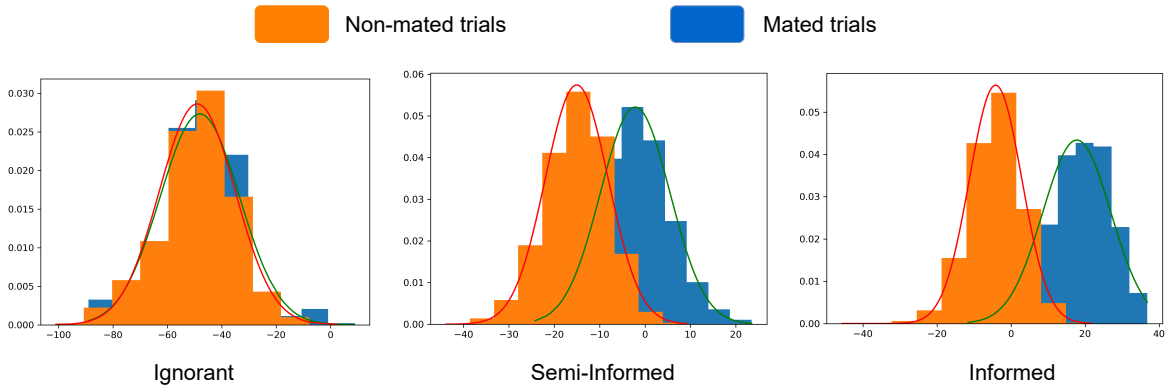


Fig. 3.6 I-vector PLDA score distribution for trials conducted on VTLN (strategy *random*) converted data by *Ignorant*, *Semi-Informed*, or *Informed* attackers. The orange distribution indicates non-mated scores, while the blue distribution indicates mated scores. The crossing between the two curves indicates the threshold for EER. More overlap means greater confusion, hence greater privacy protection.

Additional experiments were conducted to investigate the case when the attacker may not simply use the same anonymization scheme as the speaker. Based on the observations in Tables 3.3 and 3.4, a comparative study was conducted where a *Semi-Informed* attacker, trained using a particular target selection strategy, may try to deduce the speaker's identity from speech samples protected using all the other target selection strategies. Also considering the fact that the *Informed* and *Semi-Informed* attackers trained using the *random* strategy performed consistently well, a natural question arises: how well do these attackers perform on speech samples protected using the *const* and *perm* strategies?

The results, as reported in Table 3.5, indicate that the *perm* strategy may not be the best for a speaker because it is not resilient against an attacker trained using the *random* strategy. Such an attacker performs decently well against any strategy. It may be the case because having observed several different targets, the system has learned to distinguish speaker's information by ignoring the target selection and considering other discriminatory features that are not removed by switching the target identity. Hence it is crucial to discover and remove those factors that may be contributing towards the speaker's identity for complete anonymization.

Table 3.6 gives the WER obtained for each VC method, which is used as a proxy for the utility of the converted speech. Note that there is no difference between converted data in different attack scenarios, hence the WER does not depend on the attacker. VoiceMask and VTLN-based VC achieve reasonable

Table 3.5 EER (%) achieved by the i-vector based *Semi-Informed* attacker on speech samples protected with VTLN-based VC, as a function of the target selection strategy employed by the speaker and by the attacker. Bold face indicates the best attacker's strategy against each protection strategy.

Semi-Informed attacker (train + enroll)	**Protection strategy (trial)**		
	const	*perm*	*random*
const	**18.25**	26.18	25.18
perm	33.00	31.49	33.60
random	20.66	**17.35**	**18.76**

WER compared to the untransformed data, while the disentangled representation based VC produces unreasonably high WER. Note that these WERs are achieved when ASR is trained solely using converted data. In practice, many techniques can be used to optimize the WER, such as using converted data to augment clean data as will be investigated in Section 5.5.

Table 3.6 WER (%) achieved using $\text{ASR}_{\text{eval}}^{\text{anon}}$ on speech samples protected with VoiceMask, VTLN-based or disentanglement-based VC.

Subset ↓ / Strat. →	**Baseline**	**VoiceMask**	**VTLN-based VC**			**Disentangl.-based VC**		
		random	*const*	*perm*	*random*	*const*	*perm*	*random*
dev-clean	9.2	17.7	19.9	17.9	15.5	46.9	23.3	112.9
test-clean	9.4	18.1	19.8	18.4	15.9	41.5	23.7	115.1
dev-other	28.1	37.4	41.2	37.5	34.0	73.9	45.3	113.9
test-other	29.7	39.0	41.4	38.5	35.0	76.6	47.1	111.7

3.7 Experimental comparison of privacy metrics

In this section, first, the privacy metrics are compared in a simulated setting where a variety of data, representing different possible linkage scores and data points, is artificially generated to assess their response in different situations. Thereafter, the metrics are compared in a real data setting where the scores are obtained using an experimental setting similar to Section 3.6.

3.7.1 Exhibiting differences and blindspots through simulation

Two experiments are designed over simulated scores in order to exhibit the differences between the metrics. The first experiment relies on discrete scores to highlight the lack of generality of the EER. The second experiment relies on Gaussian distributed scores to exhibit the differences between $C_{\text{llr}}^{\text{min}}$ and linkability. All of the metrics are integrated in the VoicePrivacy Challenge 2020[10] and in easy-to-use open-source toolkit[11] developed by Dr. Maouche.

Discrete Scores Let us assume that there are 8 trials, i.e., pairs of utterances $p_1, \ldots, p_8$ and that the score for the i-th trial is given by the integer $LF(p_i) = i$ as shown in the header of Table 3.7. The values of EER

[10] https://www.voiceprivacychallenge.org/

[11] https://gitlab.inria.fr/magnet/anonymization_metrics

and $C_{\text{llr}}^{\min}$ vary with the label (mated vs. non-mated) of each trial. In Table 3.7, three particular cases are shown where only the labels of the last three trials (associated with scores 6, 7, and 8) change. It is observed that this has an effect on $C_{\text{llr}}^{\min}$ but not on the EER. This is because the EER searches for a single threshold of the linkage function while $C_{\text{llr}}^{\min}$ averages over all possible thresholds that the attacker might choose. Furthermore, it is also observed that the EER indicates a privacy of 25% that is half of the best achievable privacy (50%), while $C_{\text{llr}}^{\min}$ increases from half of the best achievable privacy (0.5 over 1) to higher values (0.65).

Table 3.7 $C_{\text{llr}}^{\min}$ and EER (%) with discrete scores in $\{1, \ldots, 8\}$. H (resp. $\overline{H}$) denote mated (resp. non-mated) trials.

Score	1	2	3	4	5	6	7	8	$\mathbf{C_{llr}^{min}}$	**EER**
Case 1	$\overline{H}$	$\overline{H}$	H	$\overline{H}$	H	$\overline{H}$	H	H	0.50	25.0
Case 2	$\overline{H}$	$\overline{H}$	H	$\overline{H}$	H	H	$\overline{H}$	H	0.59	25.0
Case 3	$\overline{H}$	$\overline{H}$	H	$\overline{H}$	H	H	H	$\overline{H}$	0.65	25.0

Gaussian Scores Since $D_{\leftrightarrow}^{\text{sys}}$ relies on density estimation, Gaussian distributed scores are generated to compare $D_{\leftrightarrow}^{\text{sys}}$ and $C_{\text{llr}}^{\min}$. Three different Gaussians are considered here: $G_1 \sim \mathcal{N}(1, \sigma_1)$, $G_2 \sim \mathcal{N}(2, \sigma_2)$ and $G_3 \sim \mathcal{N}(3, \sigma_3)$. Each Gaussian G_i is used to sample either mated or non-mated scores according to a key $k_i \in \{H, \overline{H}\}$. In total, four different cases are considered depending on the values of (k_1, k_2, k_3): *Mated higher* for $(\overline{H}, \overline{H}, H)$ or $(\overline{H}, H, H)$; *Mated lower* for $(H, \overline{H}, \overline{H})$ or $(H, H, \overline{H})$; *Mated in-between* for $(\overline{H}, H, \overline{H})$; *Non-mated in-between* for $(H, \overline{H}, H)$. The three given distributions are sampled in order to obtain $5,000$ mated and $5,000$ non-mated scores. Multiple standard deviations are chosen to obtain different degrees of overlap between the distributions: $(\sigma_1, \sigma_2, \sigma_3) \in \{0.1, 0.5, 1, 1.5\}^3$.

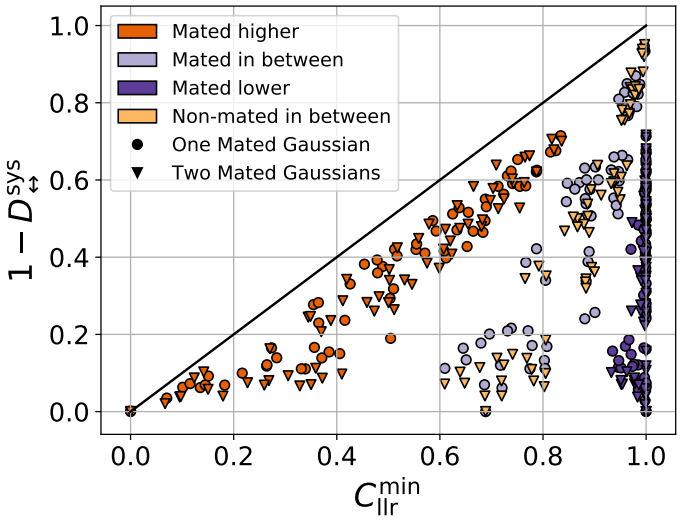


Fig. 3.7 $C_{\text{llr}}^{\min}$ vs. $1 - D_{\leftrightarrow}^{\text{sys}}$ on simulated Gaussian scores.

The results are presented in Fig. 3.7. $C_{\text{llr}}^{\min}$ and $D_{\leftrightarrow}^{\text{sys}}$ are considered equivalent when $C_{\text{llr}}^{\min}$ is equal to $1 - D_{\leftrightarrow}^{\text{sys}}$ (diagonal line). The two metrics agree to a large extent only when the mated scores are higher. When the non-mated scores are higher (mated lower), $C_{\text{llr}}^{\min}$ is always close to 1 while $D_{\leftrightarrow}^{\text{sys}}$ varies depending

on the overlap between the distributions. In the two remaining cases when the mated scores are surrounded by the non-mated scores or vice-versa, $C_{\text{llr}}^{\text{min}}$ is lower-bounded by 0.6 and the two metrics do not agree on the strength of anonymization. This is explained by the fact that threshold-based decision is meaningful in the *mated higher* case and its performance is then strongly related to the overlap between distributions, while it fails partially or totally in the three other cases.

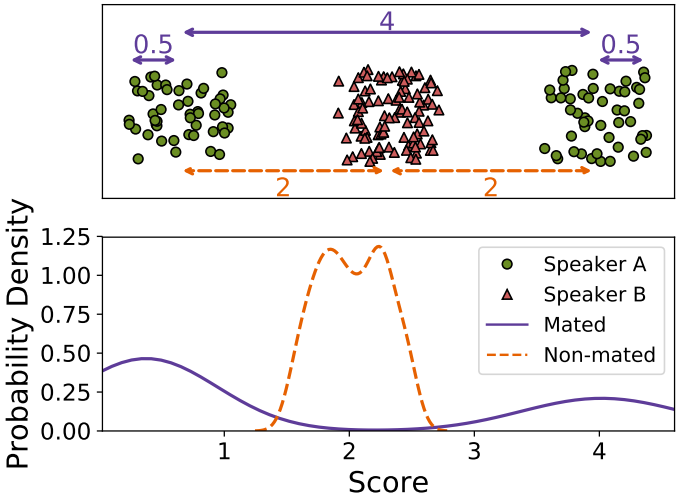


Fig. 3.8 Simulated 'non-mated in-between' data. Top: x-vectors visualized in 2D. Bottom: resulting score distributions.

To illustrate why this an issue and how this may happen in practice, in Figure 3.8, (simulated) x-vectors are drawn for multiple utterances of two speakers, which have all been anonymized by mapping them to another (target) speaker's voice. Each utterance of speaker A has been randomly mapped to the left or the right cluster, while the utterances of speaker B have been mapped to the center cluster. The resulting score distributions match the *non-mated in-between* case above. As expected, the two metrics strongly disagree: $D_{\leftrightarrow}^{\text{sys}} = 0.99$ (low privacy) and $C_{\text{llr}}^{\text{min}} = 0.81$ (high privacy). While this situation is unlikely to occur with unprocessed data (scores are then expected to match the *mated higher* case), it becomes likely once the utterances have been anonymized and the chosen anonymization method in multimodal score distributions.

3.7.2 Evaluation on real anonymized speech

In this section, the three privacy metrics are compared under real data settings. The scores and data points used for linkage attacks, along with the observations after the metric comparison are mentioned below.

Scores under consideration The linkage scores for this experiment are generated using the VC-based anonymization schemes investigated in Section 3.6. Three target selection strategies, *const*, *perm*, and *random* are used for VTLN- and disentanglement -based VC while only *random* is used for VoiceMask. For each of them, three attackers, namely *Ignorant*, *Semi-Informed*, and *Informed* attackers are considered. The attacker performs linkage attacks by computing the x-vectors of a trial utterance and an enrollment utterance and comparing them using one of three linkage functions: PLDA affinity, cosine distance, or Euclidean distance. This, along with the baseline, results in a total of 63 combinations of anonymization schemes, target selection strategies, attacker knowledge levels, and linkage functions.

Results Figures 3.9 and 3.10 compare the resulting metrics, where each dot corresponds to one of the 63 combinations above. The comparison between the EER and $C_{\text{llr}}^{\text{min}}$ (Fig. 3.9) shows a clear relation between the two metrics. In some cases the EER is stable and $C_{\text{llr}}^{\text{min}}$ varies a little bit but not significantly so. Regarding the comparison between $D_{\leftrightarrow}^{\text{sys}}$ and $C_{\text{llr}}^{\text{min}}$, a clear difference can be observed between Fig. 3.10 on real data and Fig. 3.7 on simulated Gaussian scores: on real data, the two metrics follow a clear relation.

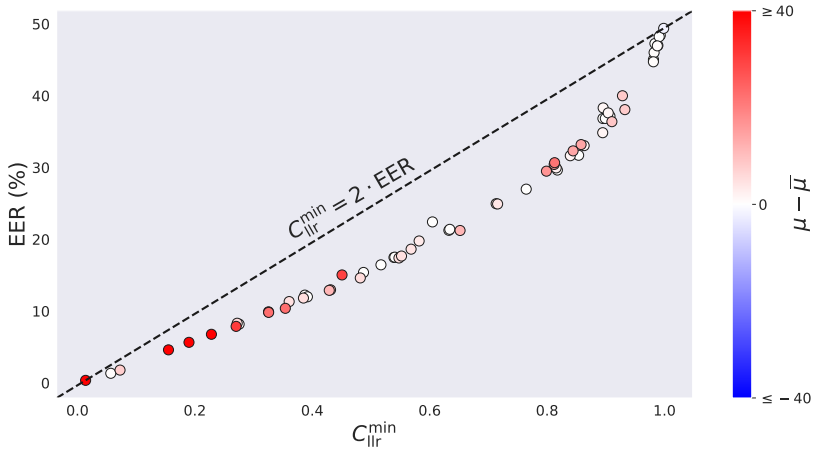


Fig. 3.9 $C_{\text{llr}}^{\text{min}}$ vs. EER (%) on real data. The color scale $\mu - \overline{\mu}$ is the difference of the means of mated and non-mated scores.

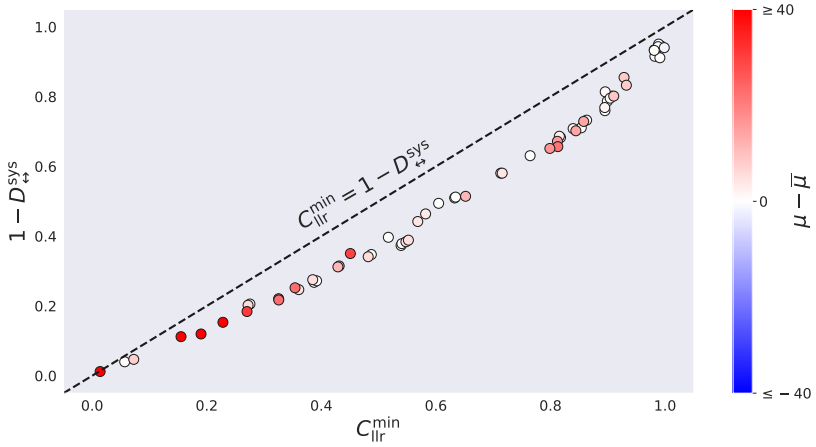


Fig. 3.10 $C_{\text{llr}}^{\text{min}}$ vs. $1 - D_{\leftrightarrow}^{\text{sys}}$ on real data. The color scale $\mu - \overline{\mu}$ is the difference of the means of mated and non-mated scores.

These results can be explained by the fact that, with few exceptions, the score distributions for the specific target selection and attack strategies considered here fall into the *mated higher* case, as can be seen from the colors associated with the dots. It is however likely that advanced target selection strategies aiming for score distributions akin to Fig. 3.7 will be developed in the near future, as these would provide an advantage against attackers making threshold-based decisions. For that reason, this study provides evidence that $D_{\leftrightarrow}^{\text{sys}}$ should be privileged as a privacy metric, since it provides very similar results to established metrics with current target selection and attack strategies, while being more robust to advanced strategies that will likely be developed soon.

3.8 Summary

In this chapter, we investigated the use of VC methods to protect the privacy of speakers by concealing their identity. Target speaker selection strategies and linkage attack scenarios based on the knowledge of attacker were formally defined. The experimental results indicated that both aspects play an important role in the strength of the protection. Simple methods such as VTLN-based VC with an appropriate target selection strategy can provide reasonable protection against linkage attacks with partial knowledge.

The characterization of strategies and attack scenarios in this chapter paves the way for designing better anonymization schemes in the following chapters. Chapter 4 generalizes the idea of *Informed* attacker to measure the amount of speaker-identifiable attributes in the intermediate representations of an ASR network. To increase the naturalness of converted speech, intra-gender VC as well as the use of a supervised phonetic classifier in VTLN can be explored. Although the adversarial learning-based approach proposed in Chapter 4 produces anonymous feature vectors instead of a speech signal as output, these vectors can be used by speech synthesis-based methods to generate a speech signal as shown in Chapter 5. Chapter 5 explores a speech synthesis based approach for generating a private speech signal, which has a high-quality and more natural output. Standard local and global unlinkability metrics [98] are used to precisely evaluate the privacy protection in various scenarios. More generally, designing a privacy-preserving transformation which induces a large overlap between mated and non-mated distributions even in the *Informed* attack scenario remains an open question which we will continue to address in the remaining chapters. In the case of disentangled representations, this calls for avoiding any leakage of private attributes into the content embeddings which can be achieved using the technique proposed in Chapter 4.

Furthermore, three metrics to assess the effectiveness of anonymization are compared: the EER, the application-independent cost function $C_{\text{llr}}^{\text{min}}$, and the linkability $D_{\leftrightarrow}^{\text{sys}}$. The EER and $C_{\text{llr}}^{\text{min}}$ assume that the attacker makes threshold-based decisions on the linkage score, while $D_{\leftrightarrow}^{\text{sys}}$ implicitly models a more powerful, non-threshold-based *oracle* attacker. The comparison on real speech data processed via three anonymization schemes with different target selection strategies and with nine attackers suggests that these metrics behave similarly. Yet, experiments on simulated data highlight fundamental differences. Specifically, the EER may yield a fixed value for situations involving different levels of privacy correctly captured by $C_{\text{llr}}^{\text{min}}$, and $C_{\text{llr}}^{\text{min}}$ becomes less informative than $D_{\leftrightarrow}^{\text{sys}}$ when the mated scores are lower or interleaved with non-mated scores. While such situations were unlikely to occur in the field of speaker verification, which involves unprocessed speech data, it is expected for them to become frequent in the field of anonymization when more advanced target selection and attack strategies are built. For this reason, this study advocates for the use of $D_{\leftrightarrow}^{\text{sys}}$ as a robust privacy metric capable of handling both current approaches and future developments in this field.

Chapter 4

Adversarial Learning based Anonymization

> For self-realization, a rebel demands a strong authority, a worthy opponent, God to his Lucifer.
>
> *Mary McCarthy*

In the previous chapter, we considered a threat model where an attacker was trying to re-identify speakers in a publicly released, anonymized speech corpus. We defined a continuum of attackers in increasing order of their knowledge about the anonymization scheme and found that the knowledgable attackers are more successful in performing the re-identification attack. This chapter considers a slightly different threat model than the one described in Section 3.1, where individuals use the speech-to-text service provided by digital assistants [183, 151]. In this context, the speech signal is sent from the user's device to a cloud-based service, as shown in Figure 4.1, where ASR and natural language understanding are performed in order to address the user request.[1] This chapter investigates if the personally identifiable information present in speech signals is *preserved* in intermediate representations constructed by an ASR network, and to what extent can these

[1] See e.g., `https://cloud.google.com/speech-to-text/`

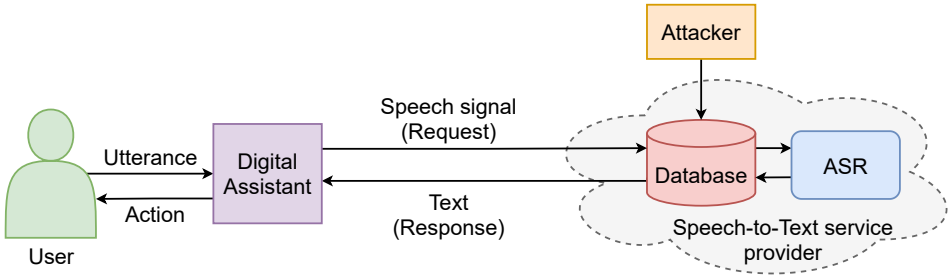


Fig. 4.1 Threat model related to speech-to-text provided by cloud-based services. The database stores raw speech signal or features rich in speaker-related information, which can be used by attackers for re-identification.

representations be used as features to re-identify the speakers within and outside the training set, which includes the users[2] of digital assistants.

The rest of the chapter is structured as follows. Section 4.1 gives a general idea of our approach where we propose an alternative ASR architecture for digital assistants. In Section 4.2, the baseline ASR model and our proposed adversarial model is described. Section 4.3 explains the experimental setup including the data sets, the ASR network architecture details and the evaluation metrics. Section 4.4 presents the obtained results and discusses their implications. Finally, Section 4.5 concludes the chapter and briefly discusses future directions.

4.1 Alternative ASR architecture

An alternative software architecture is to pre-process voice data on the device to remove some personal information before sending it to web services. Although this does not rule out all possible risks, a change of representation of the voice signal can contribute to limiting unsolicited uses of data. In this chapter, we investigate how much of a user's *identity* is encoded in speech representations built for ASR. To this end, closed- and open-set speaker recognition experiments are conducted. Recall that the *closed-set* experiment refers to a classification setting where all test speakers are known at training time. In contrast, the *open-set* experiment (a.k.a. speaker verification) aims to measure the capability of an attacker to discriminate between speakers in a more realistic setting where the test speakers are not known beforehand. The attacker is implemented with the state-of-the-art x-vector speaker recognition technique [269] as described in the previous chapter.

The representations of speech considered in this chapter are obtained from by the encoder output of end-to-end deep encoder-decoder architectures trained for ASR. Such architectures are natural in our privacy-aware context, as they correspond to encoding speech on the user device and decoding it in the cloud, also illustrated later in Figure 4.3. The baseline network [318] uses one encoder and two decoders: one based on connectionist temporal classification (CTC) and the other on an attention mechanism, briefly mentioned at the end of Section 2.2.3. Inspired by [85], the methods in this chapter propose to extend the baseline network with a *speaker-adversarial* branch so as to learn representations that perform well in ASR while hiding the speaker identity.

Several papers have recently proposed to use adversarial training for the goal of improving ASR performance by making the learned representations invariant to various conditions. While general form of acoustic variabilities have been studied [256], there is some work specifically on speaker invariance [298, 200]. Interestingly, there is no general consensus on whether it is more appropriate to use speaker classification in an adversarial or a multi-task manner, despite the fact that these two strategies implement opposite means (i.e., encouraging representations to be speaker-invariant or speaker-specific). This question was studied in [5], in which the authors conclude that both approaches only provide minor improvements in terms of ASR performance. Their speaker classification experiments also show that the baseline system already tends to learn speaker-invariant features. However, they did not run speaker verification experiments and hence did not assess the suitability of these features for the goal of anonymization.

In contrast to these studies which aim to increase ASR performance, the goal of this thesis is to assess the potential benefit of adversarial training for concealing speaker identity in the context of privacy-friendly ASR. This chapter describes the following contributions. First, CTC, attention and adversarial learning are combined within an end-to-end ASR framework. Second, a rigorous protocol is designed to quantify speaker identity in ASR representations through a series of closed-set classification and open-set verification

[2]Note that the users in this chapter are different from the ones defined in the threat model in Chapter 3, where the users were the consumers of published speech corpora. While in this chapter, the users are the customers who own and use digital assistants.

experiments. Third, as per the experiments on the Librispeech corpus [220], it is shown that this framework dramatically reduces speaker classification accuracy, but does not increase speaker verification error. Several possible reasons are suggested to explain this disparity.

4.2 Proposed model

This section starts by describing the ASR model used as a baseline, before introducing the proposed speaker-adversarial network.

4.2.1 Baseline end-to-end ASR model

The end-to-end ASR framework presented in [319] is used as the baseline architecture which is also depicted in Figure 2.8. It is composed of three sub-networks: an *encoder* which transforms the input sequence of speech feature vectors into a new representation $\mathbf{B}$, and two *decoders* that estimate the character sequence from $\mathbf{B}$. It is assumed that these three networks have already been trained using data previously collected by the service provider (which may be public data, opt-in user data, etc). Then, in the deployment phase of the system that is envisioned in this chapter, the encoder would run on the user device and the resulting representation $\mathbf{B}$ would be sent to the cloud for decoding.

The first decoder is based on CTC and the second on an attention mechanism. As argued in [319], attention works well in most cases because it does not assume conditional independence between the output labels (unlike CTC). However, it is so flexible that it allows nonsequential alignments which are undesirable in the case of ASR. Hence, CTC acts as a regularizer to prune such misaligned hypotheses. The parameters of the encoder are denoted by θ_{enc}, and those of the CTC and attention decoders by θ_{ctc} and θ_{att}, respectively. The model is trained in an end-to-end fashion by minimizing an objective function $\mathcal{L}_{\text{asr}}$ which is a combination of the losses $\mathcal{L}_{\text{ctc}}$ and $\mathcal{L}_{\text{att}}$ from both decoder branches:

$$\min_{\theta_{\text{enc}},\theta_{\text{ctc}},\theta_{\text{att}}} \mathcal{L}_{\text{asr}}(\theta_{\text{enc}},\theta_{\text{ctc}},\theta_{\text{att}}) = \beta\mathcal{L}_{\text{ctc}}(\theta_{\text{enc}},\theta_{\text{ctc}}) + (1-\beta)\mathcal{L}_{\text{att}}(\theta_{\text{enc}},\theta_{\text{att}}), \tag{4.1}$$

with $\beta \in [0,1]$ a trade-off parameter between the two decoders.

The form of the two losses $\mathcal{L}_{\text{ctc}}$ and $\mathcal{L}_{\text{att}}$ is formally described in Equations (2.20) and (2.21), respectively. Let us briefly recall the notation to eventually describe the speaker-adversarial objective. Each sample in the dataset is denoted as $\mathbf{S}_i = (\mathbf{O}_i, Y_i, z_i)$, where $\mathbf{O}_i = [\mathbf{o}_1, \dots, \mathbf{o}_T]$ is the sequence of T acoustic feature frames, $Y_i = [y_1, \dots, y_C]$ is the sequence of C characters in the transcription, and z_i is the speaker label. In the case of CTC, several intermediate label sequences of length T are created by repeating characters and inserting a special *blank* label to mark character boundaries. Let $\{\mathbf{a}_1^{(i)}, \dots, \mathbf{a}_M^{(i)}\}$ be the set of all such intermediate label sequences or alignments, where $\mathbf{a}_j^{(i)} = [\bar{y}_1, \dots, \bar{y}_T]$. The CTC loss $\mathcal{L}_{\text{ctc}}(\theta_{\text{enc}},\theta_{\text{ctc}})$ is computed as

$$\mathcal{L}_{\text{ctc}} = -\log P(Y_i|\mathbf{O}_i;\theta_{\text{enc}},\theta_{\text{ctc}}) \tag{4.2}$$

where $P(Y_i|\mathbf{O}_i;\theta_{\text{enc}},\theta_{\text{ctc}}) = \sum_{j=1}^{M} P(\mathbf{a}_j^{(i)}|\mathbf{O}_i;\theta_{\text{enc}},\theta_{\text{ctc}})$. This sum is computed by assuming conditional independence of observing a label $\bar{y}_t$ over previously observed labels $\bar{y}_{1:t-1}$, hence

$$P(\mathbf{a}_j^{(i)}|\mathbf{O}_i;\theta_{\text{enc}},\theta_{\text{ctc}}) = \prod_{t=1}^{T} P(\bar{y}_t|\mathbf{O}_i;\theta_{\text{enc}},\theta_{\text{ctc}}). \tag{4.3}$$

The attention branch does not require an intermediate label representation and conditional independence is not assumed, hence the loss is simply computed as

$$\mathcal{L}_{\text{att}}(\theta_{\text{enc}}, \theta_{\text{att}}) = -\sum_{c=1}^{C} \log P(y_c|\mathbf{O}_i, y_{1:c-1}; \theta_{\text{enc}}, \theta_{\text{att}}). \tag{4.4}$$

4.2.2 Speaker-adversarial model

In order to encourage the network to learn representations that are not only good at ASR but also hide speaker identity, we propose to extend the above architecture with what we call a *speaker-adversarial* branch. This branch models an adversary which attempts to infer the speaker identity from the encoded representation $\mathbf{B}$. We denote by θ_{spk} the parameters of the speaker-adversarial branch. Given the encoder parameters θ_{enc}, the goal of the adversary is to find θ_{spk} that minimizes the loss

$$\mathcal{L}_{\text{spk}}(\theta_{\text{enc}}, \theta_{\text{spk}}) = -\log P(z_i|\mathbf{O}_i; \theta_{\text{enc}}, \theta_{\text{spk}}). \tag{4.5}$$

Our new model is then trained in an end-to-end manner by optimizing the following min-max objective:

$$\min_{\theta_{\text{enc}}, \theta_{\text{ctc}}, \theta_{\text{att}}} \max_{\theta_{\text{spk}}} \mathcal{L}_{\text{asr}}(\theta_{\text{enc}}, \theta_{\text{ctc}}, \theta_{\text{att}}) - \lambda \mathcal{L}_{\text{spk}}(\theta_{\text{enc}}, \theta_{\text{spk}}), \tag{4.6}$$

where $\lambda \geq 0$ is a trade-off parameter between the ASR objective and the speaker-adversarial objective. The baseline network can be recovered by setting $\lambda = 0$. Note that the max part of the objective corresponds to the adversary, which controls only the speaker-adversarial parameters θ_{spk}. The goal of the speaker-adversarial branch is to act as a "good adversary" and produce useful gradients to remove the speaker identity information from the encoded representation $\mathbf{B}$. In practice, we use a *gradient reversal layer* [92] between the encoder and the speaker-adversarial branch so that the whole network can be trained end-to-end via backpropagation. Figure 4.2, which is adapted from Figure 2.14, illustrates the full architecture.

In the following, the representation computed by the encoder for a given value of λ is denoted $\mathbf{B}_\lambda$. If the encoder succeeds to produce *private representations* $\mathbf{B}_\lambda$, i.e., intermediate features that are devoid of personally identifiable attributes, then the encoder and decoder can be separately deployed on the device and in the cloud, respectively, as shown in Figure 4.3. This ensures that no personal information leaves the device, and only private attributes sufficient to decode the text content are transmitted to the central servers in the cloud. Consequently, even if the attacker gains access to the central server due to cybersecurity issues, he/she cannot link the hacked data to the original speakers, thereby protecting the anonymity of individuals using digital assistants.

The current implementation of our approach trains the whole model in an end-to-end fashion on the same machine causing the encoder and decoder to be tightly coupled with each other. If there is a change in either of their parameters, the other one must be re-trained. This limitation could be alleviated by using split learning [104] or federated learning [30] to train the model, where different parts of the ASR network can be trained on different devices using data distributed across several machines. However, such experiments are beyond the scope of this thesis.

4.3 Experimental setup

In this section, we describe the setup for experimental evaluation of the approach proposed in Section 4.2.2 in terms of privacy protection and utility. We first describe the different data sets used for training the ASR, the adversarial branch and the speaker recognition models then the architecture of the end-to-end

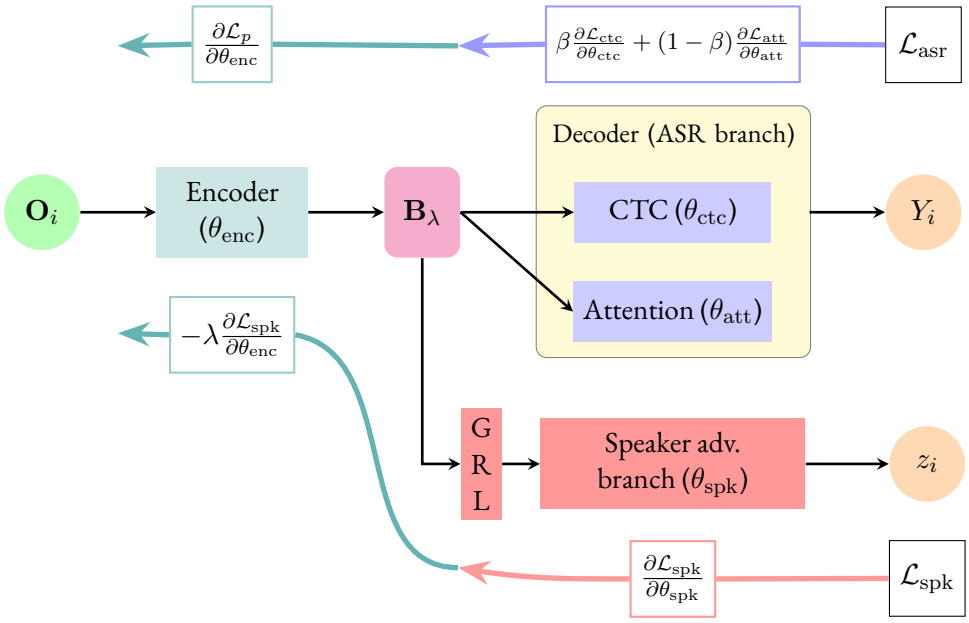


Fig. 4.2 Architecture of the proposed model. The speaker-adversarial branch is shown as a red box. The teal arrow going from GRL to encoder indicates *gradient reversal*. When the model is deployed, the encoder could reside at the client side, while the decoder can be hosted by cloud services.

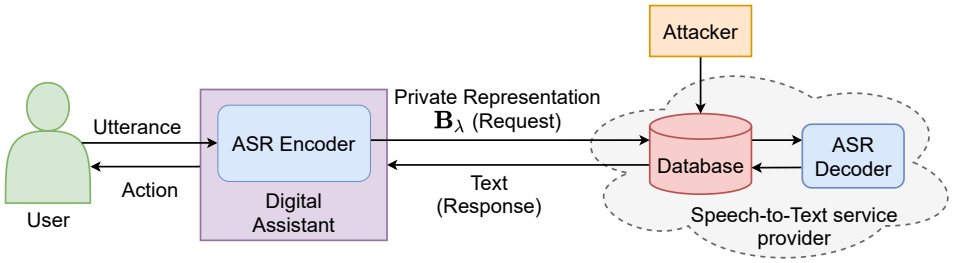


Fig. 4.3 Threat model adapted from Fig. 4.1 to reflect our proposed architecture, where the encoder resides locally in the digital assistant while the decoder is deployed on the cloud. The database stores private representations devoid of speaker-related information, hence cannot be used for re-identification

ASR network. Thereafter, we mention the training details and finally, the metrics to evaluate the privacy protection in closed- and open-set conditions are described along with the metrics for utility measurement.

4.3.1 Data sets

We use the Librispeech corpus, described in Table 4.1 and summarized in Figure 4.4 for all the experiments in this chapter. Different subsets are used for ASR training, adversarial training, and speaker verification. For the sake of clarity we refer to them as *data-full* which is used to train the ASR model and evaluate the utility, *data-adv* to fine-tune the ASR model, train the adversarial branch and evaluate the privacy in closed-set conditions, and *data-spkv* to train the speaker verification models for measuring the privacy in open-set conditions. The metrics for closed- and open-set conditions are defined in Section 4.3.4. The *data-full* set is almost the original Librispeech corpus (see Table 3.1), including *train-960* for training, *dev-clean* and

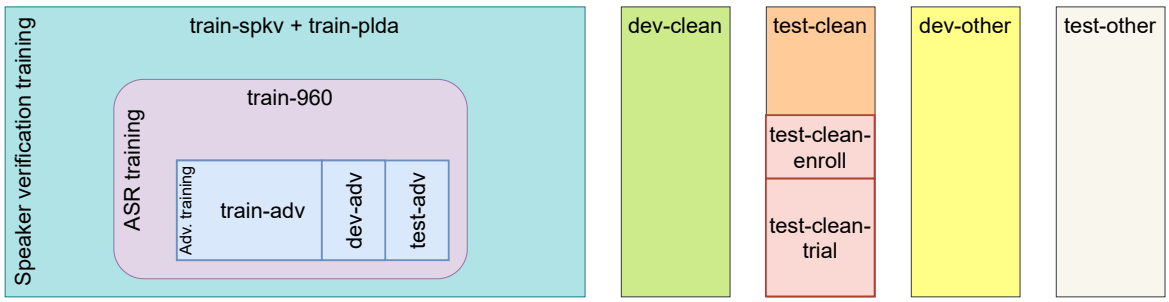


Fig. 4.4 Data set division as per Table 4.1. Each box represents a data set, and a box within it represents a subset of this data. Subsets of same color indicates that the exact same set of speakers is shared by them, and a different color indicates a disjoint set of speakers. Of course, a subset shares some of the speakers from the superset data enclosing it, but they are represented by two different colors because they are composed of non-equal sets of speakers.

Table 4.1 Splits of Librispeech used in our experiments.

dataset	data split	# utts	duration (h)
data-full	train-960	281,231	960.98
	test-clean	2,620	5.40
	dev-clean	2,703	5.39
	test-other	2,939	5.34
	dev-other	2,864	5.12
data-adv	train-adv	27,535	97.05
	dev-adv	502	1.77
	test-adv	502	1.77
data-spkv	train-spkv	373,985	1,388.79
	train-plda	422,491	1,443.96
	test-clean-enroll	438	0.75
	test-clean-trial	1496	3.60

dev-other for validation, and *test-clean* and *test-other* for test, except that utterances with more than 3,000 frames or more than 400 characters have been removed from *train-960* for faster training.

The *data-adv* set is a 100 h subset of *train-960*, which is obtained by removing long utterances from the original Librispeech *train-100* set similarly to above. It is split into three subsets in order to perform closed-set speaker identification experiments, since the speakers in the original train/dev/test splits are disjoint, i.e., comprise different set of speakers. Closed-set identification is a speaker classification task, therefore requires train/dev/test splits to contain utterances spoken by a common set of speakers. There are 251 speakers in *data-adv*: we assign 2 utterances per speaker to each of *test-adv* and *dev-adv*. The remaining utterances are used for training and referred to as *train-adv*.

For speaker verification with x-vectors [269], we use *data-spkv*, which is again derived from *data-full*. The *train-960* subset was augmented using room impulse responses, isotropic and point-source noises [157] as well as music and speech [268] as per the standard *sre16* recipe for training x-vectors [269] from the Kaldi toolkit [230], which we adapted to Librispeech. This increased the amount of data by a factor of 4. A subset

of the augmented data containing 373,985 utterances was used to train the x-vector representation and another subset containing 422,491 utterances to train the probabilistic linear discriminant analysis (PLDA) backend. These subsets are referred to as *train-spkv* and *train-plda*, respectively. For evaluation, we built an enrollment set (*test-clean-enroll*) and a trial set (*test-clean-trial*) from the *test-clean* data. Out of the 40 original speakers, 29 speakers were selected from *test-clean* based on sufficient data availability. For each speaker, we selected a 1 min subset after speech activity detection[3] for enrollment and used the rest for trials. The same evaluation protocol was used in Chapter 3, hence the details of the trials are given in Table 3.2.

4.3.2 Network architecture

For all experiments, we use the ESPnet [318] toolkit which implements the hybrid CTC/attention architecture [319]. The input features are 80-dimensional mel-scale filterbank coefficients with pitch and energy features, totalling 84 features per time frame. The *encoder* is composed of a VGG-like CNN layer followed by 5 BLSTM layers with 1,024 units. The VGG layer contains 4 convolutional layers followed by max pooling. The feature maps used in the convolution layers are of dimensions (1×64), (64×64), (64×128) and (128×128). The attention-based decoder consists of location-aware attention [39] with 10 convolutional channels of size 100 each followed by 2 LSTM layers with 1,024 units. The CTC loss is computed over several possible label sequences using dynamic programming. In all experiments, the trade-off parameter β between the two decoder losses is set to 0.5. We train a single-layer recurrent neural network language model (RNNLM) with 1,024 hidden units over the *train-960* transcriptions and use it to rescore the ASR hypotheses. The resulting WER is very close to the state of the art [333] when trained on *train-960*. Finally, we implemented the *speaker-adversarial* branch via a 3 bidirectional LSTM layers with 512 units followed by a softmax layer with 251 outputs corresponding to the 251 speakers in *data-adv*. The adversarial loss $\mathcal{L}_{\text{spk}}$ is summed across all vectors in the sequence. The speaker label z_i is duplicated to match the length of the sequence, which is smaller than T due to the subsampling performed within the encoder. Due to this subsampling as well as to the use of bidirectional LSTM layers within the encoder and the *speaker-adversarial* branch, the frame-level adversarial loss approximates well a utterance-level speaker loss that would be computed from a fixed-sized utterance-level representation, while being easier to train.

4.3.3 Training

In all experiments, we start by pre-training the ASR branch for 10 epochs over *data-full* and then the speaker-adversarial branch for 15 epochs on *data-adv* in order to get a strong adversary on the pre-trained encoded representations. Then, to reduce the computational cost all networks are fine-tuned on *data-adv* by running 15 epochs of adversarial training. Due to this, the WER (which corresponds to simple ASR training when $\lambda = 0$) is comparable to that typically achieved by end-to-end methods when trained on the *train-100* subset of Librispeech rather than the full *train-960* set. Finally, freezing the resulting encoder, we further fine-tune the speaker-adversarial branch only for 5 epochs to make sure that the reported speaker classification accuracy (ACC) reflects the performance of a well-trained adversary.

The *encoder* network contains 133.5 M parameters. To encode a 10 s audio file, it performs 1.1×10^{12} arithmetic operations which can be executed in-parallel on a 40 core CPU in 17.6 s and on a single Tesla P100 GPU in 149 ms.

[3] Speech or voice activity detection algorithms predict whether a given time frame has speech or non-speech content [285].

4.3.4 Evaluation metrics

For all tested systems, we measure ASR_{eval} performance in terms of the WER and we assess the amount of information about speaker identity in the encoded speech representation in terms of both ACC and ASV_{eval} EER. The WER is reported on the *test-clean* set. The ACC measures how well speakers can be discriminated in a closed-set setting, i.e., speakers are known at training time. The evaluation is done over the *test-adv* set using the same classifier network as the speaker-adversarial branch of the proposed model (see Section 4.2.2) after fine-tuning it for 5 epochs as mentioned in Section 4.3.3. As opposed to the ACC, the EER measures how well the representations hide the speaker identity for unknown speakers, in an open-set scenario. It reflects the process of confirming whether a person is actually who the attacker thinks he/she might be. It evaluated over the trial set (see Table 3.2) using x-vector-PLDA. The open-set evaluation is similar to the *Informed* attacker setting introduced in Section 3.1 because the ASV models are trained using the private representations (see Fig. 4.3) proposed in this chapter.

The ACC and the EER will be computed for the following representations: the baseline filterbank features (i.e., the input $\mathbf{O}_i$ in Fig. 4.2), the representations encoded by the network trained for ASR only (corresponding to $\mathbf{B}_0$) as well as those obtained with the speaker-adversarial approach (corresponding to $\mathbf{B}_\lambda$ for some values of $\lambda > 0$). The baseline measurements are obtained using the ASR_{eval} and the ASV_{eval} systems, while $\text{ASR}_{\text{eval}}^{\text{anon}}$ and $\text{ASV}_{\text{eval}}^{\text{anon}}$ systems were used to measure the performance of $\mathbf{B}_\lambda$ representations.

4.4 Results and discussion

We train our speaker-adversarial network for $\lambda \in \{0, 0.5, 2.0\}$, leading to three encoded representations $\mathbf{B}_\lambda$. Recall that $\lambda = 0$ corresponds to the baseline ASR system as it ignores the speaker-adversarial branch. Table 4.2 summarizes the results.

Table 4.2 ASR and speaker recognition results with different representations. WER (%) is reported on the *test-clean* set, ACC (%) on the *test-adv* set and EER (%) on *test-clean-trial*. The pooled scores represent the EER obtained when the male and female trials are mixed.

	Filterbank	$\mathbf{B}_0$	$\mathbf{B}_{0.5}$	$\mathbf{B}_{2.0}$
WER	–	10.9	12.5	12.5
ACC	93.1	46.3	6.4	2.5
EER Pooled	5.72	23.07	21.97	19.56
EER Male	3.34	19.38	18.26	16.26
EER Female	7.48	26.46	24.45	22.45

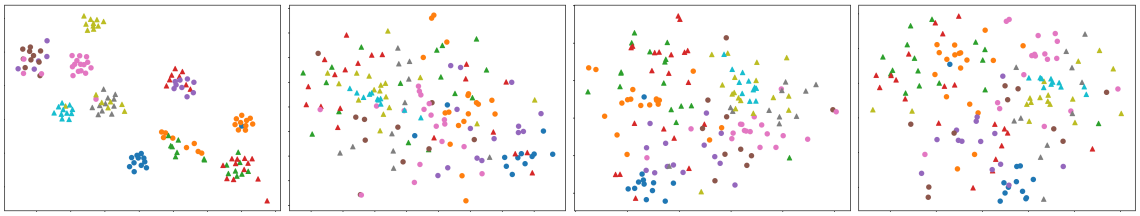
The first column presents the ACC and EER obtained with the input filerbank features, which are consistent with the numbers reported in the literature. As expected, speaker identification and verification can be addressed to very high accuracy on those features. Using the encoded representation $\mathbf{B}_0$ trained for ASR only already provides a significant privacy gain: the ACC is divided by 2 and the EER is multiplied by 4, which suggests that a reasonable amount of speaker information is removed during ASR training. Nevertheless, $\mathbf{B}_0$ still contains some speaker identity information.

More interestingly, our results clearly show that adversarial training drastically reduces the performance in speaker identification but not in verification, which is conducted in an *Informed* setting, i.e., the attacker has complete knowledge of the anonymization scheme and exploits it to train superior models. On the contrary, and counterintuitive to the speaker-invariance claims by several previous studies [298, 200, 299, 175, 328],

we observe that the verification performance actually improves after adversarial training, which implies that discrimination between the speakers became easier for an attacker. This exhibits a possible limitation in the generalization of adversarial training to unseen speakers and hence establishes the need for further investigation. The reason for the disparity between classification and verification performance might be that the speaker-adversarial branch does not inherently perform verification and hence is not optimized for that task. It might also be attributed to the representation capacity of that branch, to the number of speakers presented during adversarial training, and/or to the exact range of λ needed for generalizable anonymization. These factors of variation open several venues for future experiments.

We also notice that the WER stays reasonably low and stabilizes to 12.5% after increasing λ from 0.5 to 2. In particular, for $\lambda = 2$ the WER is just 1.6% absolute larger than the baseline ($\lambda = 0$).

We evaluate whether utterances from the same speaker stay in the same neighborhood or are scattered in the representation space. We compute t-SNE embeddings on the x-vectors extracted from 20 utterances uttered by 10 speakers (5 male, 5 female), shown in Figure 4.5. When using filterbanks, we can observe well-clustered utterances. The clusters break down when training the x-vectors on $\mathbf{B}_0$. For the x-vectors trained on $\mathbf{B}_{0.5}$ and $\mathbf{B}_{2.0}$, the clusters start to re-emerge. The silhouette scores [244] for x-vectors extracted from filterbank, $\mathbf{B}_0$, $\mathbf{B}_{0.5}$ and $\mathbf{B}_{2.0}$ representations (0.14, -0.17, -0.05 and -0.09 respectively) are consistent with the observed EER values.



(a) Filterbank (b) $\lambda = 0$ (c) $\lambda = 0.5$ (d) $\lambda = 2$

Fig. 4.5 Visualization of the x-vectors extracted from 20 utterances uttered by 10 speakers by means of t-SNE (perplexity equals to 30). Males are represented by circles and females by triangles.

4.5 Summary

We investigated the presence of speaker information in the intermediate representations of an end-to-end ASR network. The main conclusion of this investigation is that a significant amount of speaker information is present in such representations that can be used to re-identify speakers. We then propose a solution to remove the speaker-related features to produce private representations. Specifically, we proposed to combine CTC and attention losses with a speaker-adversarial loss within an end-to-end framework with the goal of learning privacy-preserving representations for ASR. Such representations could be safely transmitted to cloud services for decoding. We investigate the level of speaker anonymization achieved by adversarial training through closed-set speaker classification and open-set speaker verification metrics. Adversarial training appears to dramatically reduce the closed-set classification accuracy, seemingly indicating a high-level of anonymization. However, this observation does not match with the open-set verification results conducted in an *Informed* setting, which correspond to a strict but real scenario of a strong adversary trying to confirm the identity of a suspected speaker. Hence we conclude that adversarial training does not immediately generalize to produce anonymous representations in speech. We hypothesize that this disparity might be attributed to the representation capacity of the adversarial branch, the size of the training set, the formulation of the adversarial loss, and/or the value of the trade-off parameter with the ASR loss.

Although, in Chapter 5 for the sake of simplicity we will assume that the ASR bottleneck representations contain only linguistic features and use the untransformed representations to generate anonymized audio, in Chapter 6 we will challenge this assumption based on the findings of this chapter. It remains to be seen how well the representations generated by an adversarial network can perform in terms of privacy when they are used to generate an intelligible speech signal. As future work, we also plan to modify the speaker adversarial branch to inherently optimize for verification instead of classification and ascertain the impact of these experimental choices over different datasets, including for languages not seen in training.

Chapter 5

X-vector based Anonymization

> The measure of intelligence is the ability to change.
>
> *Albert Einstein*

As of now, we have introduced the idea of anonymization schemes that allow speakers to publish their voice data privately. We have also explained how these schemes can be reliably evaluated by simulating different attack conditions. In Chapter 3, the proposed methods produce intelligible speech signals as output, while the methods in Chapter 4 estimate private neural representations that can be directly transmitted to cloud-based services for ASR decoding. Recall that the goal of this thesis is to also preserve the utility of speech data, hence we excluded the solution of transcribing the speech and then using the output text to synthesize a waveform due the destruction of utility as explained in Section 2.2.4. It is preferred to have a waveform as the output due to the following three reasons: firstly, waveforms are easy to validate in terms of intelligibility and naturalness; secondly, due to their wide usability as published speech corpora; and thirdly, for ASR training, where the collected anonymized data must be transcribed by human annotators who listen to it. Hence, this chapter introduces an anonymization pipeline that replaces the speaker's identity in an utterance with another identity and then uses speech synthesis to generate an anonymized utterance. This pipeline was initially proposed as the primary baseline for the first VoicePrivacy challenge as described later.

This chapter is organized as follows. Section 5.1 mentions the limitations of classical voice conversion methods, and presents a contrasting approach where a flexible VC method can be built using speech synthesis tools. Section 5.2 gives an overview of the first VoicePrivacy challenge: the task definition, the data sets, the evaluation metrics, the baseline systems, and the preliminary results which lay the foundation for our upcoming contributions. In Section 5.3, we extend the primary baseline proposed in the VoicePrivacy challenge using four design choices to select the target speaker for anonymization. We empirically verify the resilience of these design choices from the perspectives of the actors involved in the anonymization process, along with the effect of pitch modification on privacy and utility. Section 5.4 further explores the privacy protection provided by the best anonymization scheme in the presence of thousands of potential speakers for re-identification, followed by the worst-case analysis which determines the empirical lower bound of such protection. Finally, in Section 5.5, we investigate a data augmentation based approach to generalize the good performance of ASR trained on anonymized corpora over original speech utterances, thereby enhancing the usability of anonymized speech. Section 5.6 summarizes the findings of this chapter and opens up the directions for the next chapter.

The research presented in Section 5.2 is a collaborative effort by the organisers of the first VoicePrivacy challenge who are listed on its website.[1] The author contributed towards the implementation of the primary baseline for the challenge. The investigations in and after Section 5.3 are essentially contributed by the author of this thesis. Nathalie Vauquier contributed towards the experiments performed in Section 5.5.

5.1 X-vector based voice conversion

Voice conversion methods are a crucial component for designing speaker anonymization schemes as discussed in Chapter 3. We briefly reviewed the different types of VC methods in Section 2.3.3 and then proposed the criteria to choose a method that is most suitable for the task of anonymization, i.e., non-parallel, many-to-many, and source/language independent. Classical VC methods such as the VTLN-based and disentangled representation based methods in Chapter 3 have addressed the "many-to-many" criterion by requiring all possible target speakers to be present in the training set, and allowing the anonymization scheme to choose one of them during deployment. In other words, the pool of target speakers is fixed and cannot be expanded without re-training. This severely restricts the capacity of anonymization schemes.

Indeed, anonymization schemes derive their strength from the amount of speaker variability that can be induced in the output speech. A large pool of speakers that could be flexibly expanded would allow an arbitrary unseen speaker to be selected as the target or even several targets to be mixed to forge an imaginary sample in speaker space, i.e., a *pseudo-speaker*.

We briefly discussed in Section 2.5 how Fang et al. [79] relaxed this limitation by introducing a VC framework based on speech synthesis that does not require the source and target speakers to be present in its training set. Figure 5.1 shows a schematic diagram of this approach. The core idea is to extract the sequences of intonation and linguistic features of the source utterance along with a single x-vector for the whole utterance, to replace that x-vector by another randomly chosen vector, and to synthesize the resulting speech.

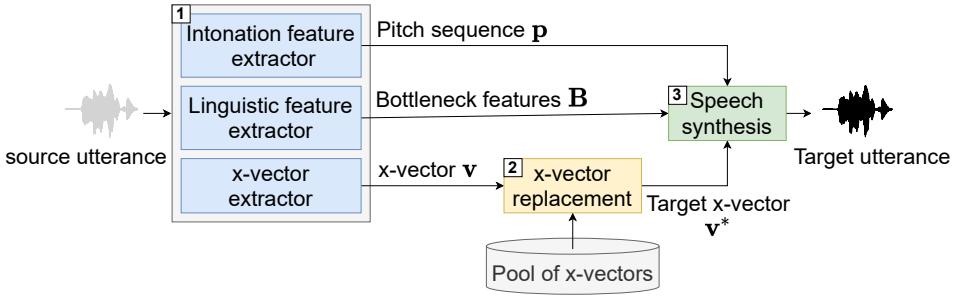


Fig. 5.1 Speech synthesis based VC framework conditioned upon a continuous speaker representation that can be replaced by unseen targets.

Every block in the given architecture is built independently using different data sets. The detailed specification of each block in this diagram is mentioned later in Table 5.6 when this architecture is used as the bedrock for the anonymization schemes. For the moment, we consider it as a framework for performing VC. Within the scope of this chapter except for Section 5.3.5, it is assumed that all the source speaker-related information is concentrated in the x-vector ($\mathbf{v}$) extracted from the utterance, and replacing it with the target speaker's x-vector ($\mathbf{v}^*$) is sufficient to remove all the identity markers of the source speaker.[2] In block 2,

[1] https://www.voiceprivacychallenge.org/

[2] This assumption is not completely true as there may be residual speaker information in other features as well. We investigate the validity of this assumption in Chapter 6.

the new target speaker x-vector $\mathbf{v}^*$ is selected from an external pool of speakers for identity replacement. It is crucial to note that the external pool can be expanded by simply adding more x-vectors into it, given that the x-vectors are extracted using the extractor in block **1**. This property allows the *flexible-pool* VC algorithm to scale up to several hundred thousand target speakers easily and independently without requiring to re-train any of its other components. The original pitch sequence $\mathbf{p}$ that represents the intonation information, the original BN features $\mathbf{B}$ representing the linguistic information, and the new x-vector $\mathbf{v}^*$ are passed to block **3**, i.e., the speech synthesis block which generates the target waveform. The speech synthesis block is described in detail in Section 2.2.4.

5.2 The first VoicePrivacy challenge

The first VoicePrivacy challenge was launched in February 2020 to introduce the objectives of the VoicePrivacy initiative [294] to the general public. It aims to promote the development of privacy preservation tools for speech technology by gathering a new community to define the tasks of interest and the evaluation methodology, and benchmarking solutions through a series of challenges. Specifically, the goals of the first challenge match the central goals of this thesis as described in Section 1.2, i.e., to develop anonymization solutions that suppress personally identifiable information contained within speech signals, and at the same time, preserve linguistic content and speech quality/naturalness. In this section, we briefly describe the task, the data sets used for training and testing, the evaluation metrics, the proposed baseline systems, and the obtained results. We limit our description to the selected parts of the challenge that are relevant to the contributions made in this thesis.

5.2.1 Anonymization task

The threat model depicted in Figure 3.1 and the actors introduced in Section 3.1, i.e., the speakers, the users, and the attackers, directly inspired the formulation of the VoicePrivacy initiative. Privacy preservation is formulated as a game between speakers who publish some data and attackers who access this data or data derived from it and wish to infer information about the speakers. To protect their privacy, the speakers publish data that contain as little personal information as possible while allowing one or more downstream goals to be achieved. To infer personal information, the attackers may use additional prior knowledge.

Focusing on speech data, a given privacy preservation scenario is specified by: (i) the nature of the data: waveform, features, etc., (ii) the information seen as personal: speaker identity, traits, spoken contents, etc., (iii) the downstream goal(s): human communication, automated processing, model training, etc., (iv) the data accessed by the attackers: one or more utterances, publicly-available data or model, etc., (v) the attackers' prior knowledge: previously published data, privacy preservation method applied, etc. Different specifications lead to different privacy preservation methods from the speakers' point of view and different attacks from the attackers' point of view.

In the context of the VoicePrivacy 2020 challenge, the following scenario is considered where each speaker passes his/her utterances through an anonymization system to hide his/her identity. The resulting anonymized utterances are referred to as *trial* data. They sound as if they had been uttered by another speaker called *pseudo-speaker*, which may be an artificial voice not corresponding to any real speaker. The task of challenge participants is to design this anonymization system. In order to allow all downstream goals to be achieved, this system should: (a) output a speech waveform, (b) hide speaker identity as much as possible, (c) distort other speech characteristics as little as possible, (d) ensure that all trial utterances from a

given speaker appear to be uttered by the same pseudo-speaker, while trial utterances from different speakers appear to be uttered by different pseudo-speakers.[3]

Attack models and evaluation The attack model and evaluation metrics for the challenge are also directly inspired from the ones described in Section 3.1 and 3.4, respectively. The attackers have access to: (a) one or more anonymized trial utterances, (b) one or more original or anonymized enrollment utterances for each speaker, (c) sometimes, the knowledge of the anonymization scheme applied by the speaker. The protection of personal information is assessed via *privacy* metrics, including objective speaker verifiability and subjective speaker verifiability. These metrics assume different attack models.

For instance, the objective speaker verifiability metrics assume that the attackers have access to a single anonymized trial utterance, several enrollment utterances, and a publicly available training corpus. Three sets of privacy metrics (see Section 5.2.3) are used for evaluating the degree of privacy protection against three attackers, i.e., *Ignorant*, *Lazy-Informed* and *Semi-Informed*. In the *Lazy-Informed* and *Semi-Informed* case, it is assumed that the trial and enrollment utterances of a given speaker have been anonymized using the same scheme, but the corresponding pseudo-speakers are different.[4] By contrast, the subjective speaker verifiability metric assumes that the attackers have access to a single anonymized trial utterance and a single original enrollment utterance.

In the following, we focus on evaluation using objective metrics. Also, for the sake of conciseness, the results are averaged over male and female speakers. Readers are referred to [295] for subjective and gender-dependent objective evaluation results.

5.2.2 Data sets

Several publicly available corpora are used for the training, development and evaluation of speaker anonymization systems. The primary anonymization system is adapted from the flexible-pool VC system introduced in Section 5.1, which consists of several blocks trained on different data sets. We first describe the data sets in this section and then the exact architecture of the anonymization scheme in Section 5.2.4. Note that the speakers in the training, development and evaluation sets are disjoint.

Training set The training set comprises the 2,800 h *VoxCeleb-1,2* speaker verification corpus [208, 44], and 600 h subsets of the *LibriSpeech* [220] and *LibriTTS* [335] corpora, which were initially designed for ASR and speech synthesis, respectively. The selected subsets are detailed in Table 5.1.

Table 5.1 Statistics of the training data sets.

Subset	Size (h)	Number of Speakers			Number of Utterances
		Female	Male	Total	
VoxCeleb-1,2	2,794	2,912	4,451	7,363	1,281,762
LibriSpeech train-clean-100	100	125	126	251	28,539
LibriSpeech train-other-500	497	564	602	1,166	148,688
LibriTTS train-clean-100	54	123	124	247	33,236
LibriTTS train-other-500	310	560	600	1,160	205,044

[3] This is akin to "pseudonymization", which replaces each speaker's identifiers by a unique key. This term is not used here, since it often refers to the distinct case when the identifiers are tabular data and the data controller stores the correspondence table linking speakers and keys.

[4] Of course, the speakers in the training set are disjoint from the enrollment set.

Development set The development set involves *LibriSpeech dev-clean* data set, which is split into trial and enrollment subsets. The speakers in the enrollment set are a subset of those in the trial set.

Table 5.2 Statistics of the development data sets.

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

Evaluation set Similarly, the evaluation set comprises *LibriSpeech test-clean* which is the same as the speaker verification trial set used in the previous chapters (also described in Table 3.2).

Table 5.3 Statistics of the evaluation data sets.

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	1,496

5.2.3 Objective metrics

Following the attack models in Section 5.2.1, objective metrics are used to assess anonymization performance in terms of speaker verifiability. We also propose objective utility metrics to assess whether the requirements in Section 5.2.1 are fulfilled. To do so an ASV system (ASV_{eval}) is trained to assess speaker verifiability and an ASR system (ASR_{eval}) is trained to assess ASR decoding error. Both systems are trained on *LibriSpeech train-clean-360* (Table 5.4) using Kaldi [230].

Table 5.4 Statistics of the training data set for the ASV_{eval} and ASR_{eval} evaluation systems.

Subset	**Size (h)**	**Number of Speakers**			**Number of Utterances**
		Female	**Male**	**Total**	
LibriSpeech train-clean-360	363.6	439	482	921	104,014

The ASV_{eval} system for *speaker verifiability* evaluation relies on the x-vector / PLDA [269] setup described in Section 2.2.5 and used in the previous chapters. Four privacy metrics that are also described in Section 3.4.1 are computed, i.e., the EER, the log-likelihood ratio costs C_{llr} and $C_{\text{llr}}^{\text{min}}$, and the linkability $D_{\leftrightarrow}^{\text{sys}}$. As shown in Fig. 3.3, these metrics are computed for four scenarios:

1. *Original*: The speaker does not perform any anonymization. The attacker uses original speech for enrollment and an ASV_{eval} system trained on original speech. This offers the lowest possible privacy protection.

2 *Ignorant*: The speaker anonymizes his/her speech, unbeknownst to the attacker who still uses original speech for enrollment and an ASV_{eval} system trained on original speech.

3 *Lazy-Informed*: The speaker anonymizes his/her speech. The attacker anonymizes the enrollment data using the same anonymization scheme as the speaker. However, he/she is not aware of the exact $\mathbf{v} \mapsto \mathbf{v}^*$ mapping from the source speaker to the pseudo-speaker. Hence, different pseudo-speakers are assigned to the trial and enrollment utterances of a given speaker.

4 *Semi-Informed* The speaker anonymizes his/her speech. The attacker anonymizes the enrollment data using the same scheme as the speaker. However, he/she is also not aware of the exact $\mathbf{v} \mapsto \mathbf{v}^*$ mapping from the source speaker to the pseudo-speaker. Hence, different pseudo-speakers are assigned to the trial and enrollment utterances of a given speaker. In addition, he/she anonymizes the training data set for the ASV_{eval} system and re-trains it to get $\text{ASV}_{\text{eval}}^{\text{anon}}$. This scenario is the one in which the speaker is most "vulnerable" despite anonymization, hence we consider it as the most trustworthy assessment of privacy.[5]

The number of mated and non-mated trials is given in Table 5.5.[6]

Table 5.5 Number of speaker verification trials for objective evaluation of speaker verifiability.

Subset		**Trials**	**Female**	**Male**	**Total**
Development	LibriSpeech dev-clean	Mated	704	644	1,348
		Non-mated	14,566	12,796	27,362
Evaluation	LibriSpeech test-clean	Mated	548	449	997
		Non-mated	11,196	9,457	20,653

The *ASR decoding error* is computed using ASR_{eval} that is based on the state-of-the-art Kaldi recipe for LibriSpeech involving a TDNN-F acoustic model (see Sec. 2.2.3) and a trigram language model. As shown in Fig. 3.5, case 1 original and case 2 anonymized trial data are decoded using the provided pretrained ASR_{eval} model and the corresponding WERs are calculated.

5.2.4 Anonymization baselines

Two different baseline systems have been developed for the challenge: (1) anonymization using x-vectors and neural waveform models, and (2) anonymization using McAdams coefficient.[7]

Baseline-1: Anonymization using x-vectors and neural waveform models The primary baseline (B1) system for the VoicePrivacy 2020 challenge depicted in Figure 5.2 is based on the flexible-pool VC method described in Section 5.1. The anonymization is performed in three steps as indicated by the three blocks, i.e., *Step* 1 extraction of x-vector $\mathbf{v}_{\text{src}}$, pitch ($\mathbf{p}$) and bottleneck ($\mathbf{B}$) features; *Step* 2 x-vector anonymization ($\mathbf{v} \mapsto \mathbf{v}^*$); *Step* 3 speech synthesis (SS) from the anonymized x-vector $\mathbf{v}^*$, and the original $\mathbf{p}$ and $\mathbf{B}$ features.

[5] The *Informed* scenario described in Section 3.1 where the attacker is aware of the exact $\mathbf{v} \mapsto \mathbf{v}^*$ mapping is not part of our study, since it falls into a security problem rather than just a privacy problem.

[6] As classically assumed in the speaker verification literature, the two speakers in each trial have the same original gender. In practice though, the gender of the original speaker may be unknown to the attacker. Hence, the resulting privacy can be seen as worst-case from the speaker's point of view and best-case from the attacker's point of view.

[7] Both baseline systems are available online: `https://github.com/Voice-Privacy-Challenge/Voice-Privacy-Challenge-2020`

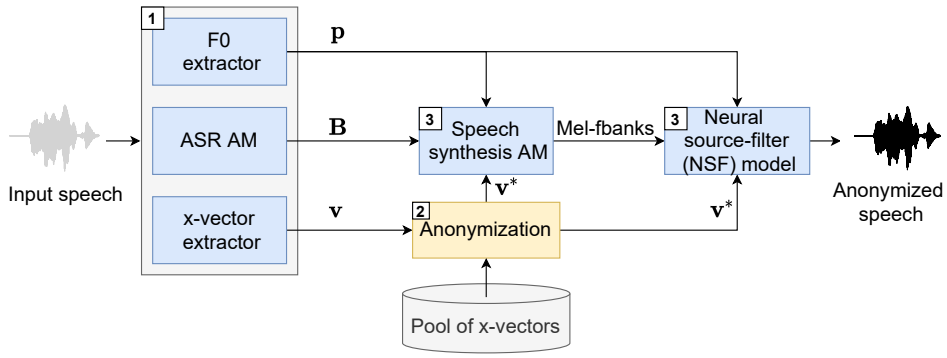


Fig. 5.2 Anonymization framework for the Baseline-1 system adapted from Figure 5.1.

The summary of each block in Figure 5.2 is provided in Table 5.6. In Step 1, to extract BN features $\mathbf{B}$, an ASR acoustic model (AM) is trained (#1 in Table 5.6). It is assumed that these BN features represent the linguistic content of the speech signal. The ASR AM has a TDNN-F model architecture that is described in Section 2.2.3, and is trained using the Kaldi toolkit [230]. To encode speaker information, an x-vector extractor with a TDNN model topology (#2 in Table 5.6) is also trained using Kaldi. In this step, $\mathbf{p}$ is estimated using the YAAPT pitch extractor as explained in Section 2.2.1.

Table 5.6 Baseline-1 system: model architectures, objective functions, output features, and training corpora. Superscript numbers represent feature dimensions.

#	Model	Description	Output features	Training data set
1	ASR AM	TDNN-F Input: MFCC^{40} + i-vectors^{100} 17 TDNN-F hidden layers Output: 6,032 tied states LF-MMI (Eq. (2.17)) and CE criteria (Eq. (2.18))	BN^{256} features extracted from the final hidden layer	Librispeech train-clean-100 train-other-500
2	X-vector extractor	TDNN Input: MFCC^{30} 7 hidden layers + 1 stats pooling layer Output: 7,232 speaker ids CE criterion	speaker x-vectors^{512}	VoxCeleb 1, 2
3	Speech synthesis AM	Autoregressive (AR) network Input: $\mathbf{p}^1$ + BN^{256} + x-vectors^{512} FF * 2 + BLSTM + AR + LSTM * 2 + highway-postnet MSE criterion	Mel-filterbanks^{80}	LibriTTS train-clean-100
4	NSF model	h-sinc-NSF in [313] Input: $\mathbf{p}^1$ + Mel-fbanks^{80} + x-vectors^{512} STFT criterion (Eq. (2.24))	speech waveform	LibriTTS train-clean-100
5	Pool of speaker x-vectors			LibriTTS train-other-500

In Step 2, for a given source speaker, a new anonymized x-vector $\mathbf{v}^*$ is computed by averaging[8] a set of *candidate x-vectors* from the speaker pool whose similarity to the x-vector of the source speaker is within a given range. The cosine distance or, optionally, the PLDA distance is used as a similarity measure The candidate x-vectors for averaging are chosen in two steps. First, for a given x-vector $\mathbf{v}$,[9] the N_w farthest candidates in the speaker pool (#5 in Table 5.6) are selected. Second, a smaller subset of N'_w x-vector candidates from this set are chosen randomly[10]. The x-vectors for the speaker pool are extracted from a disjoint data set (*LibriTTS-train-other-500*).

In Step 3, two modules are used to generate the speech waveform: a speech synthesis AM[11] that generates Mel-filterbank features given the pitch sequence $\mathbf{p}$, the anonymized x-vector $\mathbf{v}^*$, and the BN features $\mathbf{B}$; and a NSF waveform model [313] that produces a speech waveform given $\mathbf{p}$, $\mathbf{v}^*$, and the generated Mel-filterbanks. These two models are described in detail in Section 2.2.4. Both models (#3 and #4 in Table 5.6) are trained on the same corpus (*LibriTTS-train-clean-100*).

Baseline-2: Anonymization using McAdams coefficient A secondary, alternative baseline (B2) is proposed based on speech transformation which, in contrast to the primary baseline, does not require any training data. It employs the McAdams coefficient [197] to achieve anonymisation by shifting the pole positions derived from linear predictive coding (LPC) analysis of speech signals. A brief explanation of this method is given in Section 2.5. Readers are referred to [223] for more details.

5.2.5 Results

Table 5.7 reports the values of objective speaker verifiability metrics obtained before/after anonymization with Baseline-1. The EER, $C_{\text{llr}}^{\text{min}}$ and $D_{\leftrightarrow}^{\text{sys}}$ metrics behave similarly, while interpretation of C_{llr} is more challenging due to non-calibration.[12] We hence focus on the EER below. On both the development and test sets, anonymization of the trial data greatly increases the EER. This shows that the anonymization baseline effectively increases the users' privacy. The EER estimated in the *Ignorant* setting (49 to 53%), which is comparable to or above the chance value (50%), suggests that full anonymization has been achieved. However, the *Lazy-Informed* scenario results in a much lower EER (34 to 35%), which suggests that $\mathbf{p}$ and BN features retain some information about the original speaker. If the attackers have access to anonymized enrollment data, they will be able to re-identify users almost half of the time. Stricter evaluation in the *Semi-Informed* setting further reduces the EER (11 to 13%) to a closer value to the baseline and confirms the threat that the attacker can achieve significant performance gain by re-training the re-identification system with the anonymized training set.

Figure 5.3 shows a comparison between the EERs obtained when anonymization is performed using Baseline-1 and Baseline-2. It is clearly observed that for all the cases (*Ignorant*, *Lazy-Informed*, and *Semi-Informed*) case, Baseline-1 outperforms the privacy protection provided by Baseline-2.

[8] There is no guarantee that averaging produces a valid x-vector, but all our experiments show that the synthesized anonymized speech is of good quality.

[9] Following [79], we use raw x-vectors to represent speaker identity instead of x-vectors compressed and rotated by linear discriminant analysis (LDA), as classically done in the context of ASV. Unless the projected dimension is carefully chosen after several experiments, the impact of the LDA transformation on speaker-specific information cannot be ascertained. Hence we defer experiments with LDA-transformed x-vectors to a future study.

[10] In the baseline, the following parameter values are used: $N_w = 200$ and $N'_w = 100$; and PLDA was used as the distance between x-vectors.

[11] This acoustic model is distinct from the ASR acoustic model which, given an input sequence of acoustic features, extracts BN features and/or estimates the corresponding triphone posterior probabilities.

[12] In particular, $C_{\text{llr}} > 1$ is not a problem, since we care more about discrimination metrics than score calibration metrics in the first edition.

Table 5.7 Speaker verifiability achieved by the pretrained ASV_{eval} model in the original, *Ignorant* and *Lazy-Informed* scenarios, and the ASV^{anon}_{eval} model in the *Semi-Informed* case. Baseline-1 is used for anonymization. The numbers indicate the average of the results obtained over male and female trials.

Attacker	Development				Test			
	EER (%)	C^{min}_{llr}	C_{llr}	$D^{sys}_{\leftrightarrow}$	EER (%)	C^{min}_{llr}	C_{llr}	$D^{sys}_{\leftrightarrow}$
original	4.95	0.16	28.55	0.88	4.38	0.11	21.04	0.92
Ignorant	53.95	0.99	156.55	0.09	49.69	0.99	159.24	0.07
Lazy-Informed	35.47	0.88	20.53	0.23	34.43	0.87	25.10	0.24
Semi-Informed	13.17	0.40	5.25	0.68	11.46	0.35	4.05	0.69

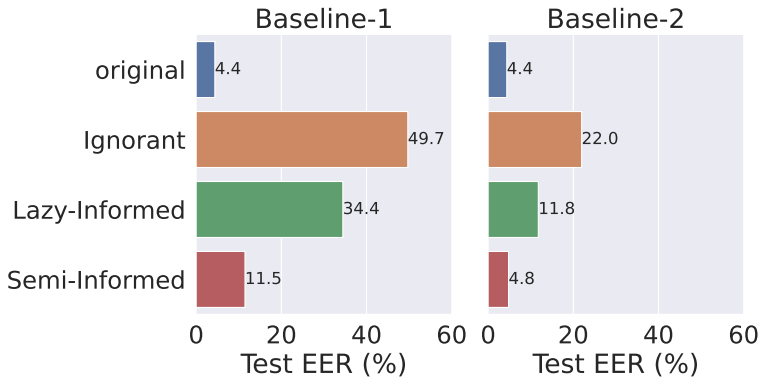


Fig. 5.3 Objective evaluation of privacy protection provided by the two baseline systems in the first VoicePrivacy challenge. Higher EER indicates better protection.

Table 5.8 reports the WER achieved before/after anonymization with Baseline-1 and Baseline-2. While the absolute WER of data sets anonymized using Baseline-1 stays below 7%, i.e., around 60% relative increase, the WER reported using Baseline-2 shows an absolute increase of 4–5%, i.e., a relative increase of more than 100%. Hence, the results achieved by Baseline-2 are inferior, both in terms of privacy as well as utility, and are detailed in [295].

Table 5.8 ASR decoding error achieved by the pretrained ASR_{eval} model. Baseline-2 is used for anonymization.

Scheme	Anonymization	Dev. WER (%)	Test WER (%)
No Anon.	original	3.83	4.15
Baseline-1	anonymized	6.39	6.73
Baseline-2	anonymized	8.77	8.88

5.3 Design choices in x-vector space

This section focuses on the explanation of the core anonymization logic implemented in the Baseline-1 algorithm, and further extends the flexibility of pseudo-speaker selection. More specifically, it aims to answer

the following questions from the speaker's and user's perspectives: Q1: *How to optimally choose and assign the target pseudo-speaker?* Q2: *How well is utility preserved?* Q3: *How much residual speaker information remains?* Furthermore, the attacker must address the following questions: Q4: *Can privacy protection be defeated using some knowledge of the anonymization scheme?* Q5: *How does the number of possible speakers affect the re-identification performance?*

To answer these questions, we extend the target pseudo-speaker generation strategy of Baseline-1 into a whole family of strategies based on four design choices. Our experiments suggest an optimal combination of design choices to balance privacy and utility (answering Q1). We train and/or evaluate ASR_{eval} and $\text{ASR}_{\text{eval}}^{\text{anon}}$ models on original and anonymized speech to assess these two forms of utility (answering Q2). We show that some speaker information remains in the pitch sequence and apply pitch transformation to remove it (answering Q3). We conduct these experiments for three types of attackers, i.e., *Ignorant*, *Lazy-Informed* and *Semi-Informed*, where stronger attackers have more knowledge about the anonymization scheme (answering Q4). Finally, we conduct additional experiments with more than 20,000 possible speakers (answering Q5).

5.3.1 Anonymization framework

In the following, we use the anonymization system shown in Fig. 5.4, that is a variant of Baseline-1 described in Section 5.2.4. Similar to Baseline-1, this system represents speaker identity, linguistic content and intonation using x-vectors ($\mathbf{v}$), BN features ($\mathbf{B}$) and pitch sequence ($\mathbf{p}$), respectively. There are two major differences as compared to Baseline-1. First, in Step 2, the mapping $\mathbf{v} \mapsto \mathbf{v}^*$ is not simply the average of the N'_w candidate x-vectors in the external pool which are farthest from $\mathbf{v}$. Instead, it is dictated by the four design choices illustrated in Figure 5.5, namely the choice of the distance metric between x-vectors, the region of x-vector space where the candidates are selected, their gender, and the assignment of the resulting target x-vector to one or all utterances of the original speaker. Second, there is a new optional Step 3 for pitch transformation which receives the pseudo-speaker target pitch statistics from the anonymization module and transforms the original pitch $\mathbf{p}$ to $\mathbf{p}^*$. Refer to Table 5.6 for details on the feature dimensions and the architectures of the models in Steps 1 and 4.

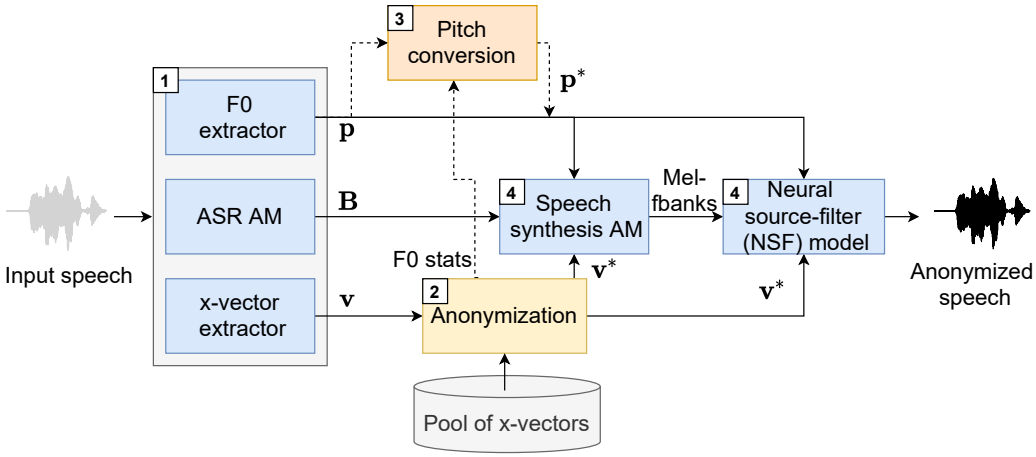


Fig. 5.4 New architecture of the anonymization system adapted from the one introduced in Section 5.2.4.

5.3.2 Proposed design choices

We now present the four design choices illustrated in Figure 5.5.

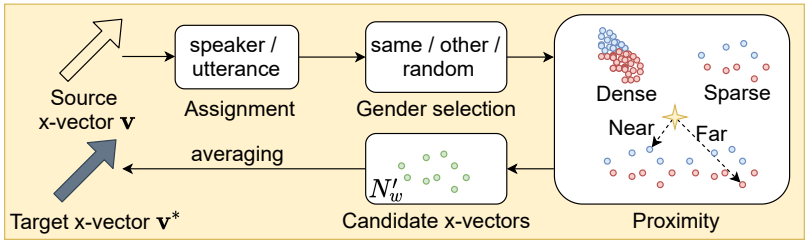


Fig. 5.5 Zoomed-in view of the x-vector anonymization step in Fig. 5.4 showing the design choices for the generation of the target x-vector.

5.3.2.1 Distance metric

To design advanced candidate selection strategies, the speaker must first choose a distance metric which dictates the properties of the x-vector space. We compare two such metrics.

The first one is the cosine distance, which was used by [79]. For a pair of x-vectors $\mathbf{v}_i$ and $\mathbf{v}_j$, it is defined as

$$d_{\cos}(\mathbf{v}_i, \mathbf{v}_j) = 1 - \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{||\mathbf{v}_i||_2 ||\mathbf{v}_j||_2}. \tag{5.1}$$

The second metric is based on PLDA [129], that is the log-likelihood ratio of the two hypotheses that $\mathbf{v}_i$ and $\mathbf{v}_j$ belong to the same speaker ($\mathcal{H}_\text{s}$) vs. different speakers ($\mathcal{H}_\text{d}$). Previous studies [150] have shown that PLDA yields state-of-the-art performance as the similarity metric between x-vectors in the context of ASV. This is attributed to its formulation which estimates the factorized within-speaker and between-speaker variability in speaker space, making it a superior metric even for short utterances [249]. The exact formulation of PLDA is given in Equation (2.26), where the parameters μ_s, R, V and D are trained on x-vectors extracted using the x-vector extractor in Step **1** from the VoxCeleb-1,2 data set that is used to train that extractor itself (see Table 5.1 for details on this data set). Hence, the log-likelihood ratio score

$$l_\text{PLDA}(\mathbf{v}_i, \mathbf{v}_j) = \log \frac{p(\mathbf{v}_i, \mathbf{v}_j | \mathcal{H}_\text{s})}{p(\mathbf{v}_i, \mathbf{v}_j | \mathcal{H}_\text{d})} \tag{5.2}$$

can be computed in closed form [242]. We propose to use $-l_\text{PLDA}$ as the "distance" between a pair of x-vectors.

5.3.2.2 Proximity

We propose three alternative criteria resulting in five different "proximity" choices to restrict the region of x-vector space from which candidate x-vectors are selected.

Random The simplest candidate x-vector selection strategy is to select N'_w x-vectors uniformly at random from the pool. Note that this strategy does not allow us to choose particular regions of interest in the x-vector space.

Far/near Alternatively, the chosen distance metric can be used to find candidate x-vectors which resemble most (*near*) or least (*far*) the original speaker $\mathbf{v}$. In essence, we rank all the x-vectors in the pool in increasing order of their distance from $\mathbf{v}$ and select either the top N_w (*near*) or the bottom N_w (*far*). To introduce some randomness, $N'_w < N_w$ x-vectors are selected out of these N_w uniformly at random.

Dense/sparse Another alternative is to identify clusters of x-vectors in the pool and rank them based on their cardinality. We construct these clusters using the Affinity Propagation [68] algorithm (see detailed procedure in Section 5.3.3.2). We filter out the cluster which is closest to the source speaker, then randomly select one cluster among those with most (*dense*) or least (*sparse*) members.[13] We then randomly select half of the members of that cluster.

In all five cases, the selected candidate x-vectors are averaged to obtain the target (pseudo-speaker) x-vector $\mathbf{v}^*$.

5.3.2.3 Gender selection

In practice, instead of applying one of these five proximity choices to the entire speaker pool, we apply it to a gender-dependent pool which consists of either all males or all females of the original pool. We propose three possible gender selection choices: *same* where all speakers in the pool have the same gender as the original speaker; *opposite* where they all have the opposite gender; and *random* where either of the two gender-dependent pools is selected at random. This allows us to avoid averaging candidate x-vectors from both genders with each other, and to assess the impact of gender selection on privacy and utility.

5.3.2.4 Assignment

The generation of the anonymized waveform is conditioned upon the x-vector sequence, whose length is equal to the number of frames in the original utterance. All the x-vectors in this sequence are identical to each other to indicate a single pseudo-speaker ($\mathbf{v}^*$) throughout the utterance. In theory, these x-vectors should also be identical across all utterances spoken by this pseudo-speaker but, according to [236], x-vectors also contain channel, duration, and phonetic information, in addition to speaker and gender. Hence, the x-vectors computed for different utterances exhibit some variations due to utterance-specific properties. To assess the effect of these variations on privacy and utility, we propose two assignment strategies for the target x-vector: speaker-level (*perm*) or utterance-level (*rand*). In the former case, we average the utterance-level x-vectors of all utterances of the original speaker into a single speaker-level x-vector $\mathbf{v}$, we generate a corresponding target x-vector $\mathbf{v}^*$, and we use it to anonymize all utterances of that speaker. In the latter case, we consider the utterance-level x-vector $\mathbf{v}_u$ for a given utterance u of the original speaker, we generate a corresponding target x-vector $\mathbf{v}_u^*$ (using the same distance metric, proximity, and gender across all utterances), and we use it to anonymize that utterance only.

5.3.3 Experimental setup

Along with the privacy evaluation using *Ignorant*, *Lazy-Informed* and *Semi-Informed* attackers that is relevant from the speakers' and attackers' perspective, the utility of ASR training is also evaluated which is relevant from the users' perspective.

5.3.3.1 Data

The experiments in Section 5.3.4 follow the VoicePrivacy Challenge setup. The training data sets for the components of the anonymization system, i.e., the ASR AM, the x-vector extractor, and the speech synthesis AM and NSF model are described in Table 5.6. The *train-other-500* subset of LibriTTS is used as the external pool of speakers for x-vector anonymization. The development and test sets are built from the *dev-clean* and

[13] Note that the terms *sparse* and *dense* do not directly reflect the density of x-vectors, since they do not take the diameter of the clusters into account. However, we find that this relation holds in practice.

test-clean subsets of LibriSpeech, respectively as described in Table 5.5. Each of these two sets consists of *trial* utterances from 40 speakers and *enrollment* utterances from a subset of 29 speakers (see Section 5.3.3.3).

5.3.3.2 Algorithm settings

The *dense* and *sparse* anonymization choices are implemented as follows. We use Affinity Propagation [68] to cluster the speakers in the external pool. This non-parametric clustering method determines the number of clusters automatically through a message passing protocol. Two parameters govern the final number of clusters: *preference* assigns prior weights to samples which may be likely candidates for centroids, and *damping factor* is a floating-point multiplier to the responsibility and availability messages. In our experiments, equal *preference* is assigned to each sample and the *damping factor* is set to 0.5. Out of 1,160 speakers in the pool, 80 clusters are found, including 46 male and 34 female. The number of speakers per cluster ranges from 6 to 36. Candidate x-vector selection is achieved by picking either the 10 clusters with least members (*sparse*) or the 10 clusters with most members (*dense*). The remaining clusters are ignored. During anonymization, one of the 10 clusters is selected at random and 50% of its members are averaged to produce the target x-vector $\mathbf{v}^*$.

5.3.3.3 Privacy evaluation

To assess the strength of anonymization against attackers with increasing knowledge, we perform the evaluation in four scenarios identical to the ones presented in Section 5.2.3, i.e., Original, *Ignorant*, *Lazy-Informed* and *Semi-Informed*.

In Section 5.3.4, privacy is assessed in terms of the *linkability* $D_{\leftrightarrow}^{\text{sys}}$ [98, 193] achieved by an x-vector-PLDA ASV system trained on the *train-clean-360* subset of LibriSpeech (anonymized $\text{ASV}_{\text{eval}}^{\text{anon}}$ in the *Semi-Informed* scenario, original ASV_{eval} otherwise). Recall that this metric computes the overlap between the distributions of PLDA scores of same-speaker and different-speaker trials as described in Section 3.4.1. It behaves similarly to the EER and $C_{\text{llr}}^{\text{min}}$ [32], but it does not rely on any restrictive assumption (e.g., threshold-based decision) which makes it a more trustworthy metric [193]. For the sake of reproducibility, we use the same set of trials as in Table 5.5. Lower linkability means higher privacy.

In Section 5.4, we also evaluate the average *rank* of the true speaker and the *top-k precision* achieved for closed-set ASI. Instead of training speaker classification systems on subsets of Common Voice, which would overfit the speakers therein, we compute the PLDA scores between each trial utterance and all enrollment utterances (one per speaker, including the true speaker) using the same x-vector and PLDA models as in Section 5.3.4 and sort them in decreasing order. The higher the rank and the lower the top-k precision, the higher the privacy.

5.3.3.4 Utility evaluation

In Section 5.3.4.1, we evaluate the utility for ASR decoding in terms of the WER achieved by an ASR_{eval} system trained on the original *train-clean-360* subset of LibriSpeech and applied to the original and anonymized utterances (case 1 and 2 in Fig. 3.5). In Section 5.3.4.2, we evaluate the utility both for ASR decoding and training in terms of the WER achieved by an ASR system trained either on the original (ASR_{eval}) or the anonymized *train-clean-360* data set ($\text{ASR}_{\text{eval}}^{\text{anon}}$) and used to decode either anonymized (case 3) or original (case 4) speech. For more details on the ASR system architecture, see Section 5.2.3. A lower WER indicates higher utility.

5.3.4 Results and discussion

The design choices introduced in Section 5.3.2 result in 54 combinations, including 48 choices corresponding to 2 distances $\times$ 4 proximities (excluding random) $\times$ 3 gender selections $\times$ 2 assignments, plus 6 choices for random proximity corresponding to 3 gender selections $\times$ 2 assignments. To assess the impact of these choices, our experiments are organized according to the three actors in our threat model. First, the speaker finds the **two** most promising combinations of design choices on the development set in terms of privacy in the *Ignorant* and *Lazy-Informed* scenarios and utility for ASR decoding. This is motivated by the high computational cost of anonymizing the *train-clean-360* subset of LibriSpeech and retraining ASV and ASR systems on it, which prevents the evaluation of privacy in the *Semi-Informed* scenario and utility for ASR training for all 54 combinations. Second, the user assesses the utility of these two combinations for both ASR training and decoding. Third, the attacker quantifies the resulting privacy in the *Semi-Informed* scenario, which leads us to identify the best combination among them. Finally, we show how the proposed pitch transformation further improves privacy in this scenario with some loss of utility.

5.3.4.1 Speaker's perspective

We first evaluate the design choices from the speaker's perspective in terms of privacy in the *Ignorant* and *Lazy-Informed* scenarios and utility for ASR decoding on the development set. The results are displayed in the form of swarm plots, i.e., scatter plots where each dot represents the privacy or utility value associated with one combination of design choices. In order to avoid overlapping dots with similar values, the dots are spread horizontally.

Distance Figure 5.6 evaluates the effect of the chosen distance metric on privacy. We observe that both cosine distance and PLDA result in similarly low linkability in the *Ignorant* case but PLDA marginally outperforms cosine distance (i.e., it results in a lower linkability) in the *Lazy-Informed* case. Since both distance measures perform similarly in terms of utility (see Fig. 5.11(a)), PLDA has an advantage. Therefore we consider only PLDA as the distance metric in the following experiments.

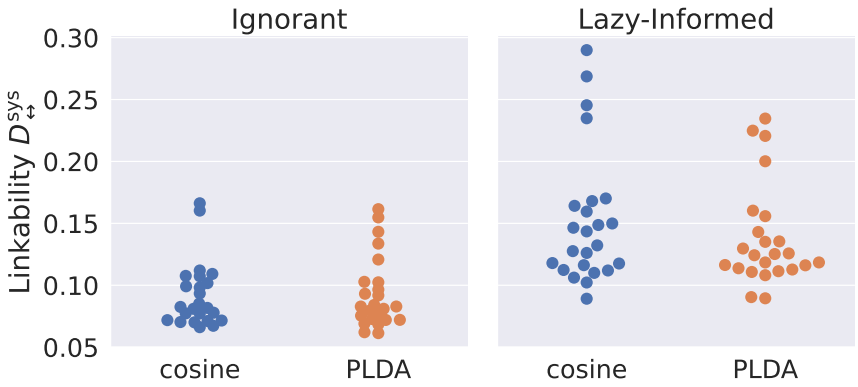


Fig. 5.6 Privacy against *Ignorant* and *Lazy-Informed* attackers depending on the distance choice. Each swarm plot shows the 24 linkability values on the development set resulting from all combinations of proximity (excluding *random*), gender selection, and assignment choices.

Proximity Next, we assess the five choices of target *proximity* described in Section 5.3.2.2, namely *random*, *near*, *far*, *sparse* and *dense*. The distance metric is fixed to PLDA and the values of N_w and N'_w are fixed to

200 and 100, respectively.[14] We discover the clusters in x-vector space and select *pseudo-speakers* from *sparse* and *dense* clusters using the procedure described in Section 5.3.3.2.

We observe in Fig. 5.7 that, although selecting candidate x-vectors *far* from the original speaker achieves the lowest linkability in the *Ignorant* case together with the *random* strategy, it is largely outperformed in the *Lazy-Informed* case by selection from *sparse* or *dense* clusters and by the *random* strategy. This shows that clustering based pseudo-speaker mapping results in more robust anonymization as compared to simple distance-based mapping.

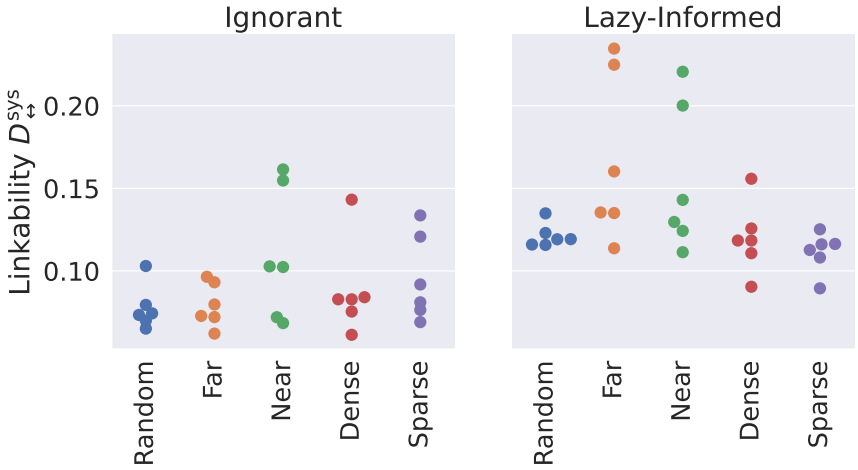


Fig. 5.7 Privacy against *Ignorant* and *Lazy-Informed* attackers depending on the proximity choice. Distance is fixed to PLDA. Each swarm plot shows the 6 linkability values on the development set resulting from all combinations of gender selection and assignment choices.

Compared to the *sparse* selection strategy, the *dense* strategy provides comparable privacy protection in the *Lazy-Informed* case, but much higher utility (see Fig. 5.11(b)). This can be attributed to the fact that speakers in *sparse* clusters stand out more from the crowd than those in *dense* clusters, therefore they are more likely to suffer from poor ASR performance.

Finally, *random* target selection yields similar privacy protection in the *Lazy-Informed* case and slightly better utility as compared to *dense*. Hence we consider the *random* and *dense* strategies to be the best choices for proximity.

Gender selection We now investigate the gender selection strategy described in Section 5.3.2.3. The distance is fixed to PLDA and proximity to *dense* or *random*. As per the results shown in Fig. 5.8 it is hard to find the best choice for gender selection in terms of privacy since the linkability is not consistently lower for any specific choice.

In order to make a suitable choice, we introduce the additional requirement that the chosen anonymization scheme obfuscates the original speaker's gender. The different anonymization schemes can be visually compared in Fig. 5.9. Same gender selection (Fig. 5.9 (b)) causes male and female clusters to move apart. A similar result is observed with opposite gender selection (not shown in the figure). On the contrary, *random* gender-selection (Fig. 5.9(c) and 5.9(d)) results in a non-separable boundary between genders.

Furthermore, we conduct gender identification experiments over the original and anonymized x-vectors shown in Fig. 5.9 to measure the degree of gender obfuscation caused by *same* vs. *random* gender selection.

[14] We noticed a sharp decline in utility for smaller values of N'_w.

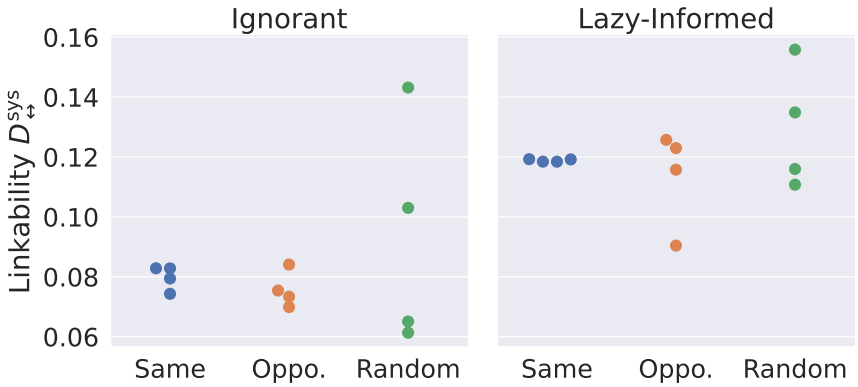


Fig. 5.8 Privacy against *Ignorant* and *Lazy-Informed* attackers depending on the gender selection choice. Distance is fixed to PLDA and proximity to *dense* or *random*. Each swarm plot shows the 4 linkability values on the development set resulting from the assignment choice and the 2 proximity choices.

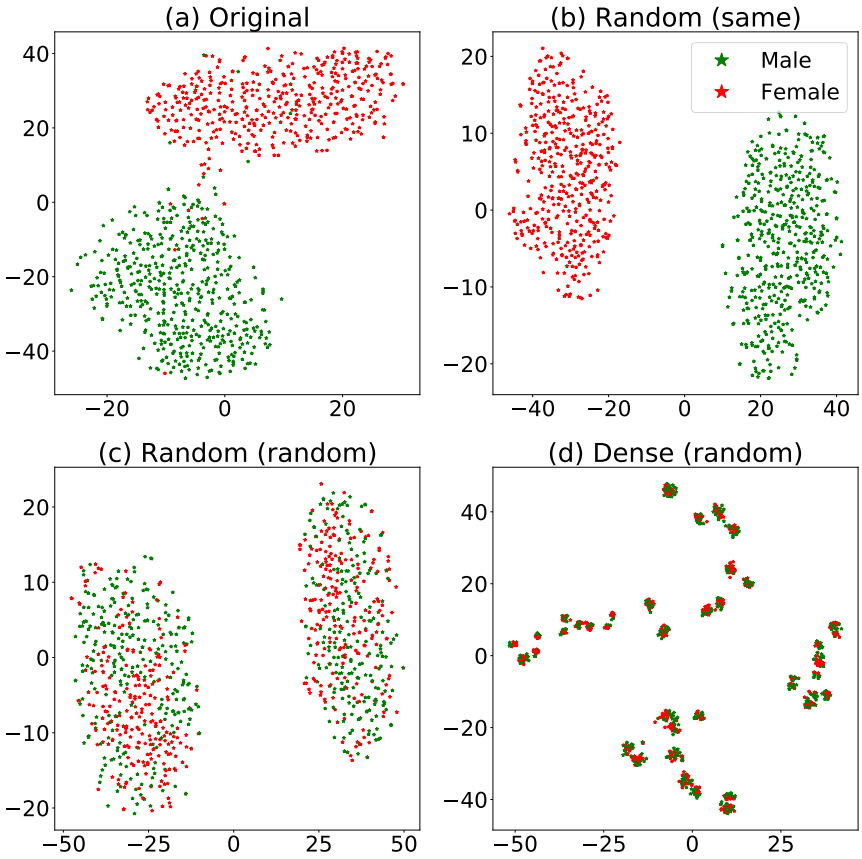


Fig. 5.9 t-SNE visualization of speaker-level x-vectors from the LibriSpeech *train-clean-360* data set transformed using different proximity (*random* or *dense*) and gender selection (*same* or *random*, in parentheses) choices. Gaussian pitch normalization (see Section 5.3.5) has been used in all the three cases.

We employ the k-nearest neighbour algorithm with 5-fold cross-validation to predict the gender of speakers in the LibriSpeech *train-clean-360* data set which contains 921 speakers. The mean cross-validation accuracy

for each data set reported in Table 5.9 corroborates the visual observations above. Therefore we consider the random strategy to be the best choice for gender selection.

Table 5.9 Gender identification accuracy over original and anonymized x-vectors extracted from LibriSpeech *train-clean-360*.

Anonymization scheme	**Mean cross-validation accuracy (%)**
Original	98.58
Random (same)	100.00
Random (random)	70.46
Dense (random)	53.31

Assignment Finally the design choice of *assignment* is examined from the speaker's perspective as described in Section 5.3.2.4. The distance is fixed to PLDA, proximity to *dense* and gender selection to *random*. The results reported in Fig. 5.10 show that *utterance-level* pseudo-speaker assignment results in lower linkability as compared to *speaker-level* assignment.

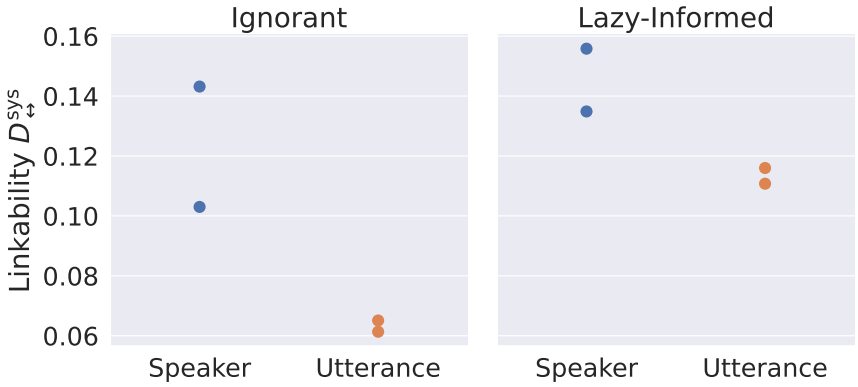


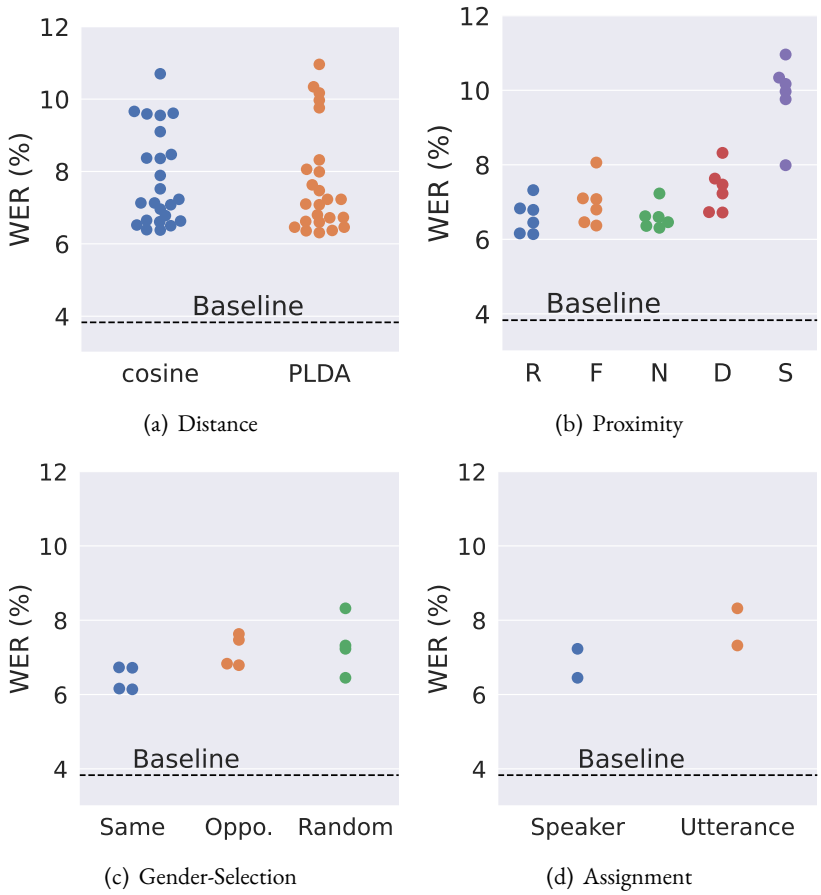
Fig. 5.10 Privacy against *Ignorant* and *Lazy-Informed* attackers depending on the assignment choice. Distance is fixed to PLDA, proximity to *dense* or *random*, and gender selection to *random*. Each swarm plot shows the 2 linkability values on the development set resulting from the 2 proximity choices.

The WER resulting from *utterance-level* assignment is higher than from speaker-level assignment (see Fig. 5.11(d)). However, in order to conform with the four requirements of the anonymization task mentioned in Section 3.2 of the VoicePrivacy Challenge evaluation plan [293], we propose to use *speaker-level* assignment. This ensures that all utterances from a given original speaker appear to be uttered by the same pseudo-speaker.

Based on these indications, the speaker may choose specific parameters according to their application needs. For the sake of further experimentation, we choose distance as **PLDA**, proximity as **random** or **dense**, gender selection as **random** and assignment as **speaker-level** to be the best combinations of design choices based on our observations.

5.3.4.2 User's perspective

We now present some complementary results from user's perspective where we measure the feasibility of using the anonymized speech corpus in downstream tasks such as ASR training. Recall that in our threat



(a) Distance (b) Proximity

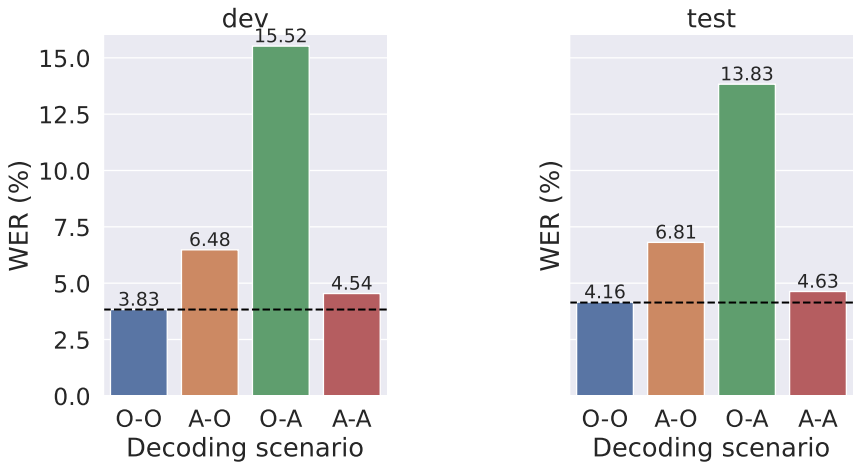
(c) Gender-Selection (d) Assignment

Fig. 5.11 Utility of anonymized speech in terms of WER compared to the original (baseline) speech depending on the different design choices. Each swarm plot shows the WER values on the development set for each gender and for a given design choice. The remaining design choices are fixed in the same way as in Figs. 5.6, 5.7, 5.8 and 5.10.

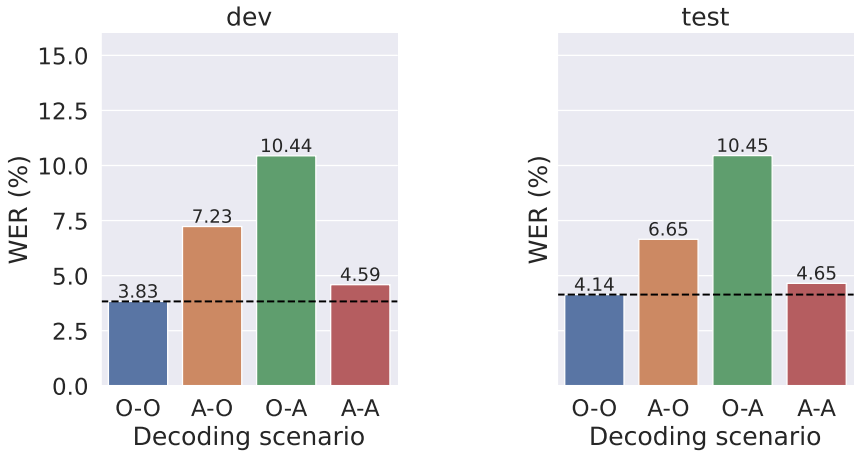
model, the user is an actor who consumes the anonymized speech corpus for some specific application. The primary concern of any user is the quality of speech in terms of naturalness and intelligibility and its usefulness in downstream tasks. We specifically exhibit the quality of anonymized speech in terms of its viability to train a good ASR_{eval} model. In Section 5.3.5 we will investigate how naturalness and intelligibility can be increased using pitch interpolation techniques.

Figure 5.12 shows the results of the two anonymization methods. The four bars in each plot represent the four decoding scenarios mentioned in Figure 3.5: O-O indicates original (non-anonymized) speech being decoded by the ASR_{eval} model trained on non-anonymized speech (case 1), A-O indicates anonymized speech being decoded by the same ASR_{eval} model (case 2), O-A indicates original speech being decoded by the $\text{ASR}_{\text{eval}}^{\text{anon}}$ model re-trained on anonymized speech (case 4), and A-A indicates anonymized speech being decoded by the same $\text{ASR}_{\text{eval}}^{\text{anon}}$ model (case 3).

We observe that the A-A (red) bar is almost always equal to the O-O (blue) bar. This indicates that the two proximity choices produce viable speech corpora for training the ASR model with a WER as low as the baseline. The middle two bars indicate a mismatch between training and decoding data. The WER



(a) Random proximity



(b) Dense proximity

Fig. 5.12 Performance of $\text{ASR}_{\text{eval}}^{\text{anon}}$ models re-trained on anonymized speech obtained using *random* or *dense* proximity. Distance is fixed to PLDA, gender selection to *random*, and assignment to *speaker-level*.

degradation is much higher when original speech is decoded using the re-trained model (case 4, O-A) than when anonymized samples are decoded using the original model (case 2, A-O). Such asymmetry indicates a "loss of generalization" when ASR is trained using anonymized speech, due to the unintentional exclusion of certain factors of variability of the original speech.

In conclusion, we have shown that the anonymized speech data is suitable for training a viable ASR acoustic model with very little loss of generalization, provided that decoding is also conducted on anonymized data. This requirement is not a problem for offline ASR decoding (e.g., for movie subtitling), but it becomes a challenge for online ASR decoding (e.g., in voice assistants), due to the computational cost of the the current anonymization pipeline and the fact that it processes entire utterances in batch mode. In order to overcome that challenge, we explore a way to improve the ASR performance when decoding on non-anonymized data in Section 5.5.

5.3.4.3 Attacker's perspective

The primary objective of the attacker is to deduce the original speaker's identity from the anonymized speech, i.e. to achieve high linkability while conducting speaker authentication trials.

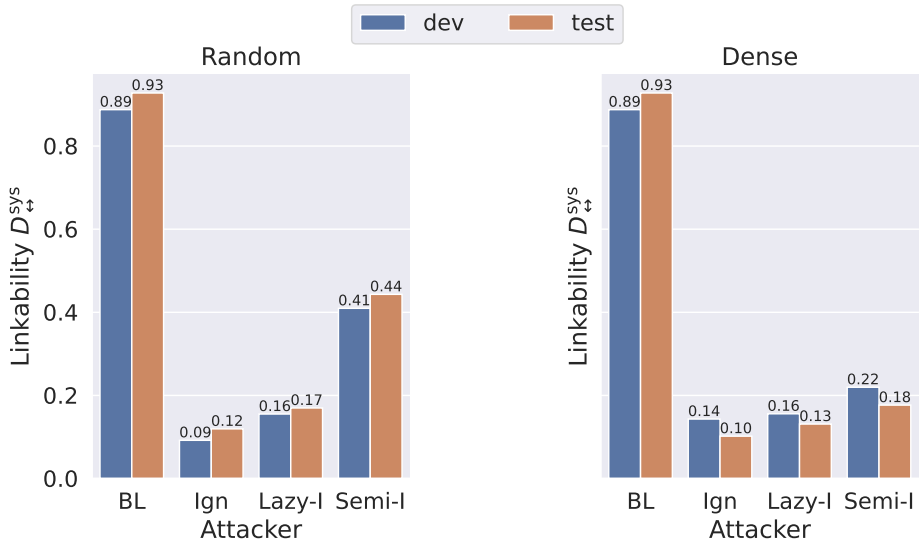


Fig. 5.13 Performance of ASV_{eval}^{anon} models re-trained on anonymized speech obtained using random or dense proximity. Distance is fixed to PLDA, gender selection to *random*, and assignment to *speaker-level*. BL = Original (baseline), Ign = *Ignorant*, Lazy-I = *Lazy-Informed* and Semi-I = *Semi-Informed* attacker.

The results for the two proximity choices and the four attack scenarios are shown in Fig. 5.13. We observe that the linkability increases gradually as we move from the *Ignorant* to the *Semi-Informed* attacker. It goes up to 0.44 for *random*, but stays below 0.22 for *dense* even in the strongest scenario. This indicates the robustness of *dense* proximity over *random*. Therefore, we ultimately recommend the following combination of choices to the speaker: **PLDA** distance, **dense** proximity, **random** gender-selection, and **speaker-level** assignment. We recall that the latter choice is a requirement set by the VoicePrivacy challenge task. Whenever *speaker-level* assignment is not required, we recommend *utterance-level* assignment for higher privacy.

Our experiments exhibited the robustness of the selected design choices against an attacker who has complete knowledge of the anonymization scheme and its parameters and only lacks the knowledge of exact pseudo-speaker targets. As opposed to signal processing based methods, such as Baseline-2, which provide no protection against a strong attacker, the design choices selected for the x-vector based method are capable of cutting the attackers' linkability down to half, or even a quarter of the baseline value.

5.3.5 Pitch conversion

Until now, we have assumed that among the three sets of features extracted by the anonymization framework, namely the BN features $\mathbf{B}$, the original x-vector $\mathbf{v}$, and the pitch contour $\mathbf{p}$, only the original x-vector $\mathbf{v}$ is transformed into the target pseudo-speaker $\mathbf{v}^*$ while the other two are left unchanged. Yet, the intonation features of an utterance contribute towards the speaker's identity and the presence of the original $\mathbf{p}$ might reveal some information about the speaker [83]. Also, keeping the pitch sequence $\mathbf{p}$ unchanged while possibly changing the gender of the x-vector results in inconsistent features which may affect the naturalness of the synthesized speech. Hence, we employ pitch conversion to better conceal the identity as well as to enhance the naturalness of the output speech.

We use logarithm Gaussian pitch normalization [182], where the original non-zero[15] values of $\mathbf{p}$ are linearly interpolated to the target $\mathbf{p}$ in the logarithmic domain using the mean and standard deviation of the original speaker's and the pseudo-speaker's pitch sequences as already described in Equation (3.3). The term $\mathbf{p}_{\text{tgt}}$ in the given equation is used as the converted pitch, $\mathbf{p}^*$. The statistics for the pseudo-speaker are computed by aggregating the pitch sequences of all the utterances of all the target speakers composing the pseudo-speaker, i.e., $\mathbf{p}_{\text{ps}}$. These statistics are stored during the x-vector anonymization (block **2** in Figure 5.4) and passed to the pitch conversion module (block **3**) during speech synthesis. A similar method is also employed by Champion et al. [37] in the same setting to study the effect of pitch conversion on privacy and utility for different genders.

Additionally, we propose two more methods, i.e., *percentile* and *minmax* based pitch conversion. The percentile pitch conversion is based on mapping the specific percentile of the original pitch distribution to the corresponding percentile of the target pitch distribution. Let $\mathbf{p}_{\text{ps}}^{\texttt{sorted}}$ be the aggregated, non-zero pitch sequence of all the utterances of all the speakers that compose the pseudo-speaker $\mathbf{v}^*$, sorted in ascending order. Given $\mathbf{p}_{\text{ps}}^{\texttt{sorted}}$, the pitch conversion is achieved as follows. First, the non-zero pitch values $\mathbf{p}[i]$ from the source utterance can be converted into percentile values $\varrho[i]$ using

$$\varrho[i] = \frac{\texttt{rank of}\ \mathbf{p}[i]\ \texttt{in}\ \mathbf{p}^{\text{sorted}}}{\texttt{length}(\mathbf{p}^{\text{sorted}})} \times 100, \tag{5.3}$$

where $\mathbf{p}$ represents the sequence of non-zero pitch values, and $\varrho[i]$ is the percentile of $\mathbf{p}[i]$ in the sorted pitch sequence of the source utterance, $\mathbf{p}^{\text{sorted}}$. Then, the converted pitch values $\mathbf{p}^*[i]$ corresponding to each $\varrho[i]$ are selected from $\mathbf{p}_{\text{ps}}^{\text{sorted}}$:

$$\mathbf{p}^*[i] = \mathbf{p}_{\text{ps}}^{\texttt{sorted}} \left[\left\lfloor \frac{\texttt{length}(\mathbf{p}_{\text{ps}}^{\texttt{sorted}}) \times \varrho[i]}{100} \right\rfloor \right]. \tag{5.4}$$

Such a mapping between the two pitch sequences can be considered an instance of one-dimensional optimal transport between the two distributions [308].

Finally, we describe the minmax pitch conversion as the linear scaling of source pitch based on the range of target pitch values. The source pitch values are first transformed into the $[0, 1]$ range, i.e., the minimum and maximum source pitch values would be 0 and 1, respectively. Thereafter, each of these values are scaled to match the target pitch range depending on the minimum and maximum value of the pseudo-speaker pitch sequence. The resulting pitch values can be simply computed as

$$\mathbf{p}^*[i] = (\mathbf{p}[i] - \min(\mathbf{p})) \times \frac{\max(\mathbf{p}_{\text{ps}}) - \min(\mathbf{p}_{\text{ps}})}{\max(\mathbf{p}) - \min(\mathbf{p})} + \min(\mathbf{p}_{\text{ps}}). \tag{5.5}$$

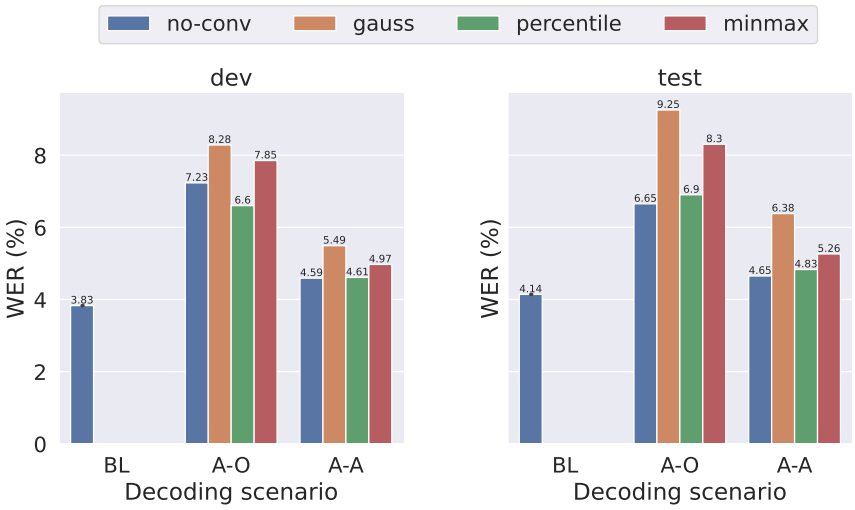
Note that the percentile and minmax pitch conversion are performed on raw pitch values in Hz.

The advantage of percentile or minmax based conversion is that all the resulting values come from the set of valid pitch values, while in case of Gaussian normalization, the computed pitch might not be within the valid range of pitch values. To the best of our knowledge, percentile-based pitch conversion is a novel approach which has not been previously reported in the literature.

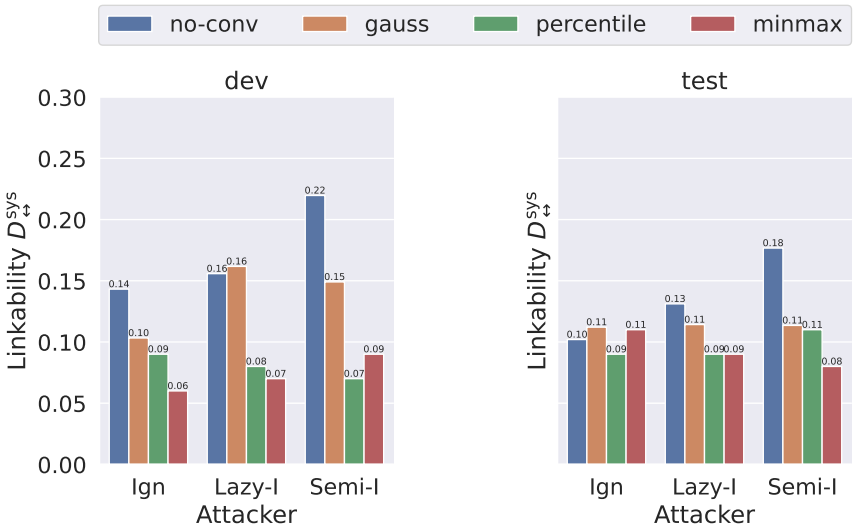
It is observed in Fig. 5.14(a) that Gaussian normalization and to a lesser extent minmax scaling of pitch significantly increase the WER (thereby causing a loss of intelligibility), while percentile-based transformation maintains the original WER. Figure 5.14(b) shows that all the three methods substantially reduce the

[15] The zeros correspond to the unvoiced/silence regions in speech. We assume that the removal of zeros does not affect privacy, since these values are equal to zero for all speakers and therefore do not convey identity information.

linkability, especially in the *Semi-Informed* case. This implies that pitch conversion removes some of the residual speaker information in the anonymized speech, thereby improving privacy protection. Again, the percentile based method performs better than Gaussian and minmax pitch conversion over the development set, while the minmax method outperforms the others on the test set. Due to better performance in terms of the WER, percentile-based pitch conversion can be proposed as a suitable method for privacy protection. It is also worth mentioning that the naturalness of cross-gender voice conversion noticeably improves after percentile-based pitch conversion, according to informal listening. A preliminary investigation reveals that the some loss of intelligibility is attributable to the Gaussian interpolation technique which introduces some illegal values in the transformed pitch sequence that do not belong within the range of pitch expected by the speech synthesis module. This is remedied by choosing percentile based mapping between the source and



(a) $\text{ASR}_{\text{eval}}^{\text{anon}}$ performance



(b) $\text{ASV}_{\text{eval}}^{\text{anon}}$ performance

Fig. 5.14 Performance of $\text{ASR}_{\text{eval}}^{\text{anon}}$ and $\text{ASV}_{\text{eval}}^{\text{anon}}$ after pitch conversion as compared to original pitch.

target sequence which ensures that only valid pitch values are selected as the new values. We do not further explore this direction in this thesis and use Gaussian normalization for pitch conversion in the next chapter.

5.4 Large-scale speaker study

In this section we analyze the attacker's performance as a function of the number of speakers in the enrollment set, i.e., the number of speakers by which the trial utterance to be re-identified could possibly have been uttered. This number depends on the attacker's prior knowledge since a smaller number of speakers reflects the ability of the attacker to narrow down the search to a smaller number of suspects using contextual information.[16] Our main goal is to study whether the speaker's identity can be hidden in the crowd or can still be revealed to some extent by ASV or ASI within a large enrollment speaker population.

To do so, we employ Mozilla's Common Voice data set because of its large number of speakers. The data set is described in Table 5.10, and to the best of our knowledge this is the first time it is used for ASV and privacy related experiments. We increase the enrollment speaker population exponentially and measure the attacker's performance at each step.

Previous research by Sholokhov et al. [262, 261] studies a similar phenomenon from a voice spoofing perspective where an attacker desires to be accepted through an ASV authentication system by finding the "closest impostor" who would be accounted as a false alarm. The attacker has access to a speech sample of a target speaker and the scoring mechanism of the ASV system. They show that the chance of acceptance of the impostor may reach up to 50% in the worst case as the population approaches 10^5 impostors. Another similar problem is posed by the Multi-target speaker detection challenge [263] where membership (TOP-S) and identification (TOP-1) of a speaker must be assessed from a large set of blacklisted speakers. They show that the performance in both cases gradually degrades as the number of speakers in the blacklist increases. In the following, we do not consider only the "closest impostor" like [261], and test the overall linkability of speakers as the potential non-mated trials increase multifold.

5.4.1 Data

In this section, we employ the same trained models and the same external pool of speakers as described in Section 5.3.3, but we build multiple test sets from the Mozilla Common Voice [10] English corpus, in order to study the attacker's success against anonymization with a larger number of possible speakers. This corpus contains more than 52,000 speakers, out of which we select up to 24,610 male speakers using gender identification (see Section A.1). The details of Common Voice enrollment and trial sets are given in Table 5.10.

Table 5.10 Statistics for the Mozilla Common Voice enrollment and trial sets.

Subset		#
CV-enroll	Number of speakers	24,610
	Number of utterances	320,085
CV-trial	Number of speakers	20
	Number of utterances	4,696
	Number of mated trials	4,696
	Number of non-mated trials	115,563,864

[16] Attacker may obtain this contextual information by inspecting the metadata/statistics of the public, anonymized data set, or by simply listening to individual utterances.

Remark 5.1 *We notice that the WER obtained over the Common Voice trial set (see Table 5.10) using the ASR_{eval} model trained on the original LibriSpeech train-clean-360 data set increases from the baseline* 4.71% *(O-O) to* 12.64% *(A-O) after anonymization. This looks reasonable given that ASR_{eval} is trained on the original LibriSpeech training set.*

5.4.2 Privacy evaluation metrics

To assess the performance of a re-identification attack, we first report the results in terms of binary decision metrics namely, EER and linkability as reported in the previous sections. Thereafter we perform closet-set ASI and report performance in terms of the *rank* and *top-k* membership metrics. We compute the PLDA similarity score for a particular speaker with all the speakers in the selected population (which always contains the true speaker) and sort it in descending order. Ideally the true speaker must rank first but if anonymization is successful his/her rank will increase. The rank and top-k metrics help us assess the relative difficulty of ASI in the presence of a large number of speakers. Since adding more speakers to the population is expected to increase the rank of the true speaker, we also report the normalized rank, that is the absolute rank divided by the number of speakers in the population.

We also plot the *chance-level* rank that is the expected rank of the true speaker when the attacker is selecting it at random among all the speakers. This rank can be computed as follows. Let $R \in \{1, \ldots, N_{\text{spk}}\}$ be the set of all possible ranks for a given speaker that can be obtained with probability $P(R)$. Then the expected rank is equal to:

$$\mathbb{E}(R) = \sum_{R=1}^{N_{\text{spk}}} R \cdot P(R). \tag{5.6}$$

To obtain the chance-level rank, we set $P(R) = \frac{1}{N_{\text{spk}}}$ Hence the chance-level rank is given by

$$\begin{aligned} \mathbb{E}(R) &= \frac{1}{N_{\text{spk}}} \sum_{R=1}^{N_{\text{spk}}} R \\ &= \frac{1}{N_{\text{spk}}} \frac{N_{\text{spk}}(N_{\text{spk}}+1)}{2} \\ &= \frac{N_{\text{spk}}+1}{2}. \end{aligned} \tag{5.7}$$

When the rank is normalized, we divide the chance-level rank by N_{spk} to obtain the normalized chance-level rank

$$\frac{N_{\text{spk}}+1}{2N_{\text{spk}}} \approx 0.5. \tag{5.8}$$

5.4.3 Experimental setup

As mentioned before, we select 24,610 speakers (see Table 5.10) as the total population possessed by the attacker from the set of male speakers whose total duration is more than 10 s after removing silent frames using voice activity detection (VAD). The maximum duration for each speaker in the population is limited to 2 min. Only those utterances whose Signal-to-Noise ratio (SNR) is greater than 75 dB are selected. We compute the SNR using the WADA-SNR [152] algorithm.[17] Out of these, 20 speakers whose total duration is greater than 5 min are selected for testing, which represents the publicly released data subjected

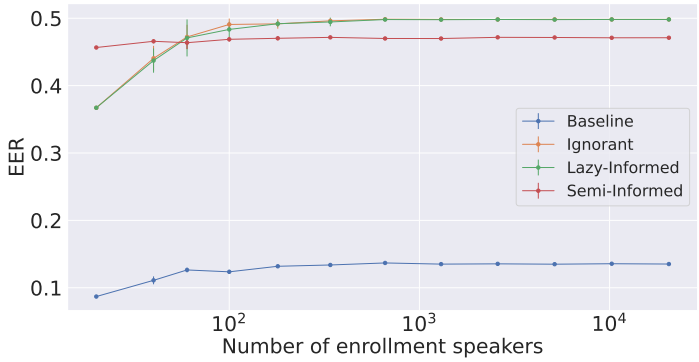
[17] https://gist.github.com/johnmeade/d8d2c67b87cda95cd253f55c21387e75

to re-identification attack. We manually listened to each speaker in the test set to confirm that it is a distinct male speaker. After computing PLDA scores between the enrollment speaker population and the test speakers, we get 4,696 mated scores and 115,563,864 non-mated scores.

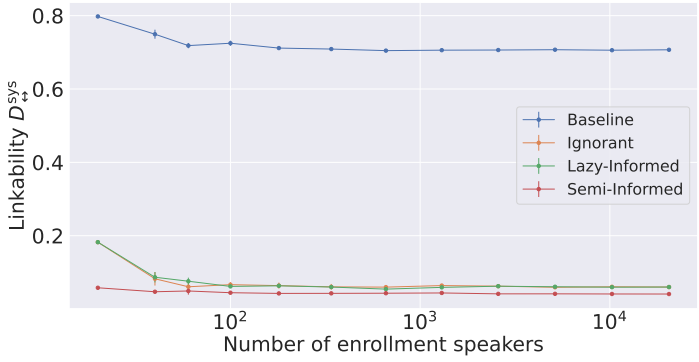
Initially we select the subset of scores which are computed only among the 20 speakers in the test set. Thereafter we double the number of other speakers in the population at each step, i.e., the total number of speakers increases from 20 to 20,500, and include the scores corresponding to these speakers. The newly added speakers are randomly sampled 5 times from the entire enrollment speaker population to avoid any bias.

5.4.4 Average-case analysis

Let us first discuss the ASV measures as shown in Fig. 5.15. We notice in Fig. 5.15(a) that the baseline EER starts with a value of 7% and slightly increases to 12% as the number of speakers increases. The three attackers perform much more poorly. The EER achieved by the *Ignorant* and *Lazy-Informed* attackers starts from 37% and quickly increases to reach 50% when the number of speakers exceeds 660. The EER curve for the *Semi-Informed* attacker follows a similar shape, although it remains below those of the *Ignorant* and *Lazy-Informed* attackers. Linkability follows a similar trend except that the *Semi-Informed* attacker displays the lowest value at all steps.



(a) Equal Error Rate

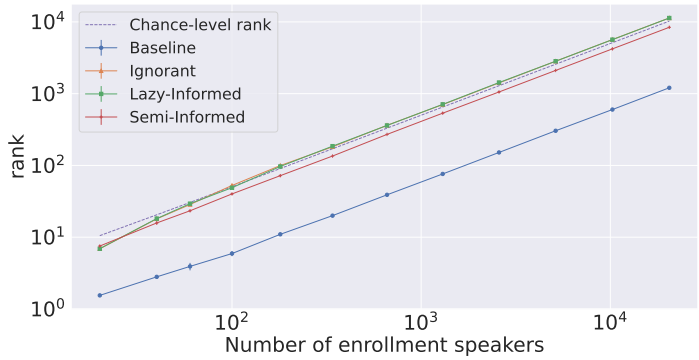


(b) Linkability

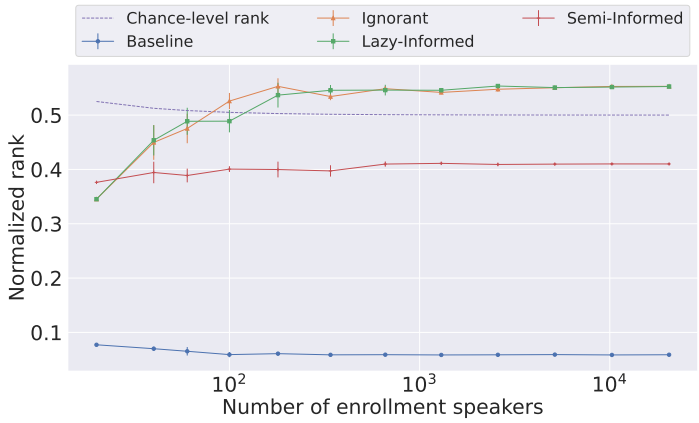
Fig. 5.15 Open-set ASV performance of different attackers in terms of EER and linkability as a function of the number of speakers in the population.

Figure 5.16 shows the un-normalized and normalized rank of the true speaker obtained by different attackers before and after anonymization. In Fig. 5.16(a), we notice a steep rise in the value of absolute

rank, i.e. a decline in the ASI performance, for original as well as anonymized data. However the baseline performance is well below the chance-level rank even with thousands of speakers in the enrollment set, which indicates the distinctive characteristics of speakers in the population. All the attackers start with better performance than chance-level but soon converge very close to the chance-level rank as the number of speakers increases. We also plot the normalized rank and the normalized chance-level rank in Fig. 5.16(b). We observe that this plot resembles the EER performance depicted in Fig. 5.15(a). The ASI performance obtained by the *Ignorant* and *Lazy-Informed* attackers quickly degrades and converges to a value worse than the normalized chance-level rank, while the *Semi-Informed* attacker maintains a consistent performance, which is slightly better than chance-level.



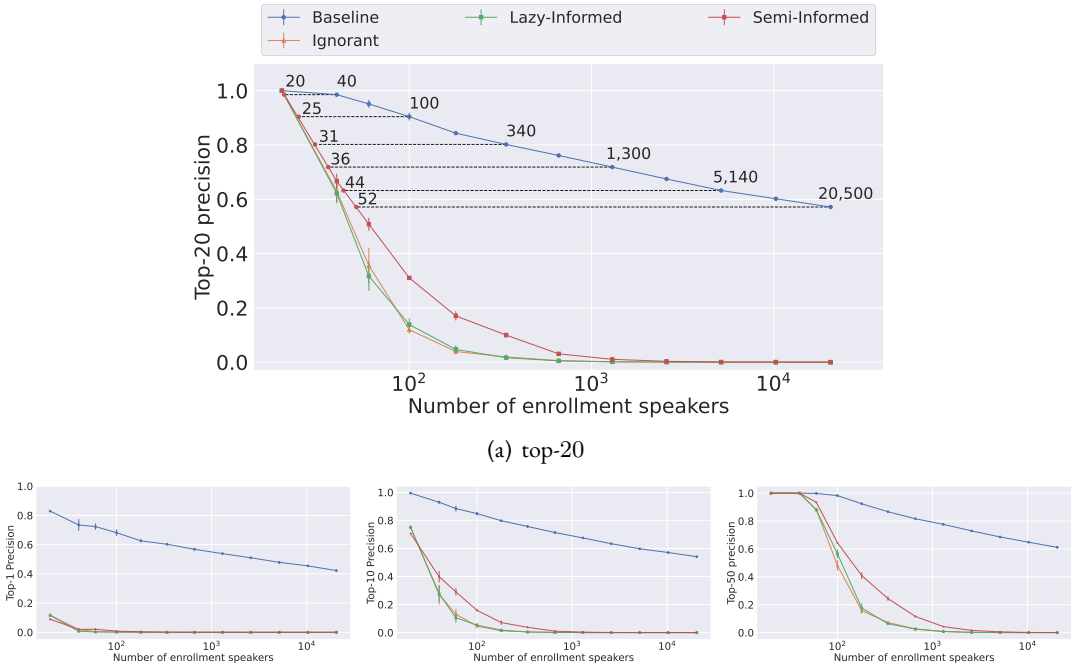
(a) Rank



(b) Normalized rank

Fig. 5.16 Closed-set ASI performance in terms of un-normalized and normalized rank obtained by different attackers as a function of the number of speakers in the population.

We further study the top-k precision obtained by different attackers and compare it to the baseline performance. We obtain the precision for four different values of k, i.e., 1, 10, 20 and 50. We focus mainly on $k = 20$ because we assume that an attacker will realistically look at the top-20 results when he/she wants to shortlist the probable speaker identities. We observe in Fig. 5.17 that the precision drops much faster after anonymization as compared to the baseline, i.e., hiding the identity of an *anonymized* speaker in a crowd of n speakers is equivalent to hiding the *original* speaker in a crowd of N speakers, where N increases at a much faster rate than n. Note that the "crowd" here refers to the enrollment speaker data set possessed by the attacker.



(a) top-20

(b) top-1 (c) top-10 (d) top-50

Fig. 5.17 Top-k precision of ASI for different attackers as a function of the number of speakers in the population, for $k = 1$, $k = 10$, $k = 20$, and $k = 50$. The numbers of speakers needed before anonymization (N on the blue curve) and after anonymization (n on the red curve) to achieve an equivalent drop in precision are highlighted in the top-20 precision curve.

We notice that without anonymization the speakers can be uniquely identified, i.e., as rank 1 with 40% accuracy among 20,000 speakers (Figure 5.17(b)), whereas there is a negligible chance of being recognized uniquely after anonymization. The chance of being recognized improves with and without anonymization as we increase the sphere of shortlisted speakers (k). For $k = 10$ (Figure 5.17(c)), the protection provided by 700 enrollment speakers in original case is similar to the protection provided by just 20 enrollment speakers after anonymization. Concretely, we can infer from Fig. 5.17(a) that while the attacker's ability to re-identify the speaker naturally reduces with the number of candidate speakers, the best instance of our anonymization scheme with 50 candidates speakers guarantees the same anonymization level as raw speech with 20,000 speakers.

5.4.5 Worst-case analysis

In the previous section, especially in Figure 5.16(b) we observed the normalized rank of original and anonymized utterance averaged over all trial speakers. A lower normalized rank increases the average risk of re-identification. Yet, some speakers may be easier to re-identify than others. To ensure optimal protection, speaker and data publishers would be interested in answering the following questions related to the worst case: *What is the distribution of normalized ranks after anonymization as compared to the original speech?*

To answer the questions above, we analyze the results in terms of the normalized rank U^{worst} of the worst-performing utterance in the data set, the normalized rank S^{worst} of all utterances from the worst-performing speaker averaged over these utterances, and the normalized rank U_S^{worst} of the single worst-performing

utterance from each speaker averaged over all speakers. We only consider the ranks obtained using the original speech (baseline) and the *Semi-Informed* attacker here.

First, we plot the distribution of normalized ranks of all the trials in the baseline case in Figure 5.18, where each subplot represents the number of speakers in the enrollment set. As expected, a majority of utterances exhibit a normalized rank very close to zero and hence are extremely vulnerable to re-identification attacks even in the presence of thousands of enrollment speakers. The values of U^{worst}, S^{worst}, and U_S^{worst}, represented by the red, green and black dashed lines, respectively, are all close to zero. In fact, they overlap each other. The goal of anonymization is to increase these values.

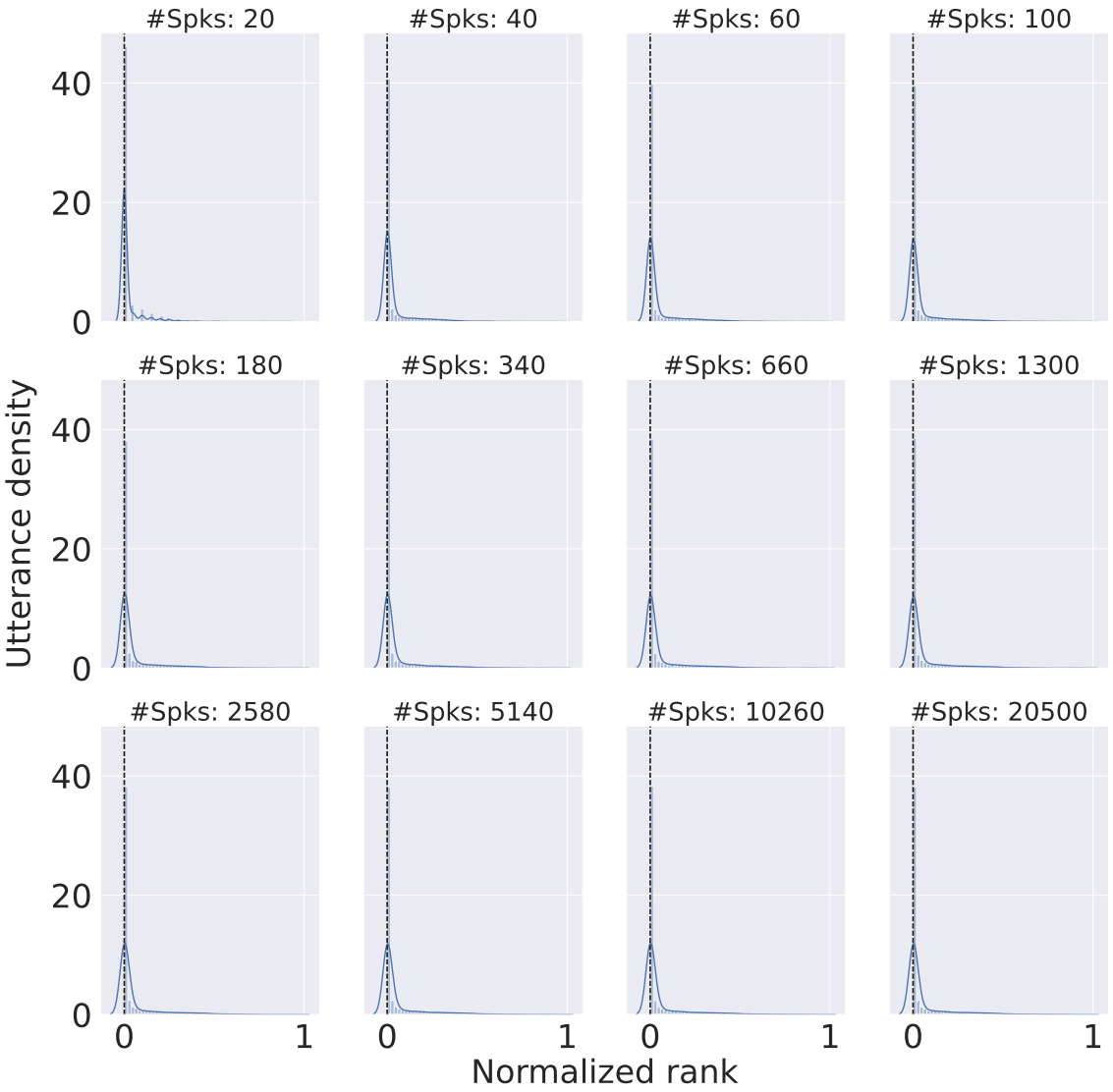


Fig. 5.18 Distribution density of the normalized rank in the baseline case as a function of the number of enrollment speakers. The dashed vertical lines show the values of U^{worst} (red), S^{worst} (green), U_S^{worst} (black).

Figure 5.19 shows the normalized rank distribution in the *Semi-Informed* setting. Although several utterances still tend to exhibit low ranks, the utterance density becomes more evenly distributed over the $[0, 1]$ range, thereby protecting most utterances from re-identification attacks. Moreover, U^{worst} is much

worse than S^{worst} and U_S^{worst}, indicating that not all utterances of a single speaker are vulnerable. The gap between S^{worst} and U_S^{worst} indicates that a majority of speakers have their worst-performing utterances better protected than the overall worst-performing utterance and the worst-performing speaker. The normalized rank of the worst performing utterance, the worst performing speaker, and the single worst-performing utterances from all the speakers is described in Appendix A.2.

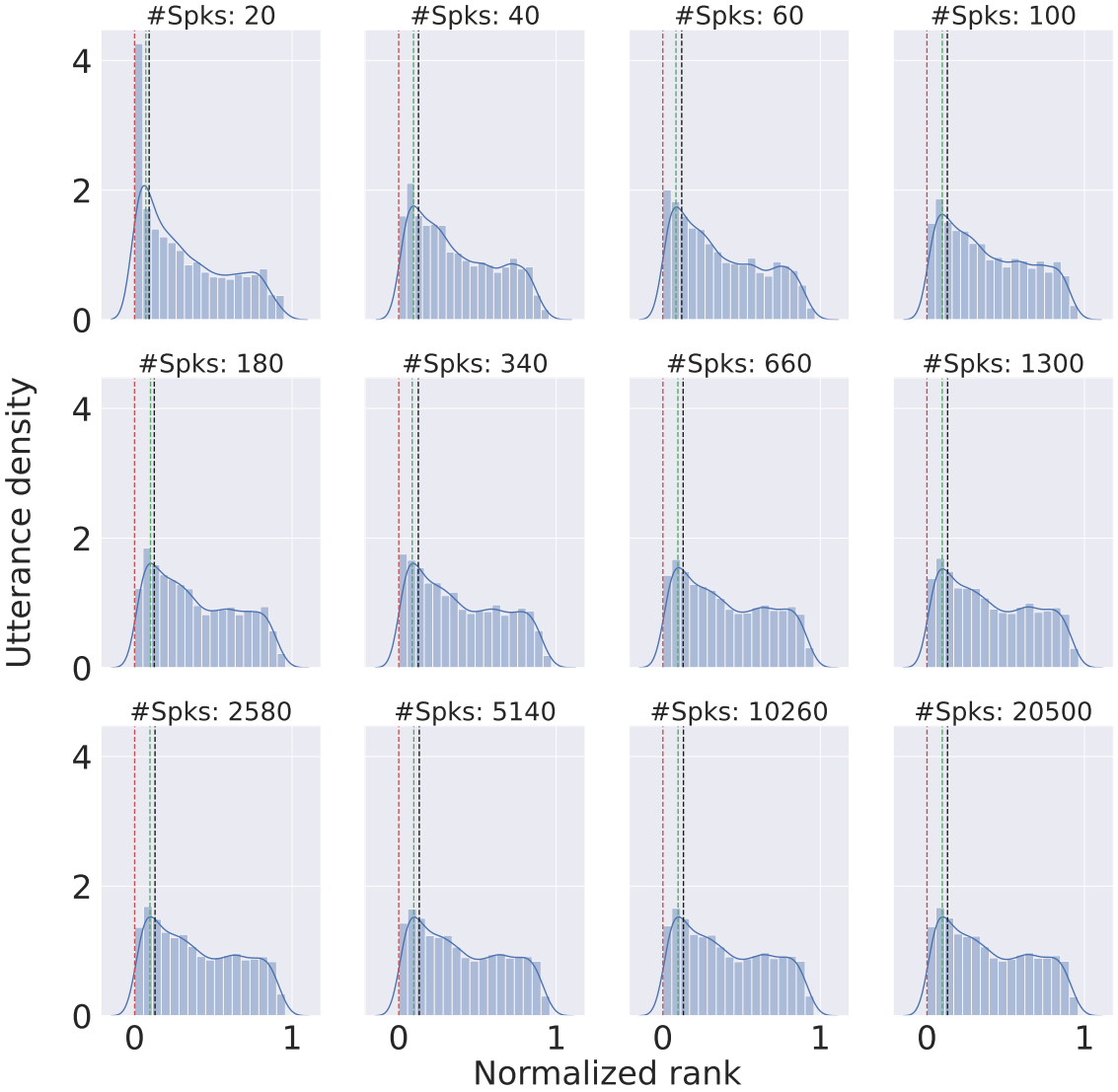


Fig. 5.19 Distribution density of the normalized rank in the *Semi-Informed* case as a function of the number of enrollment speakers. The dashed vertical lines show the values of U^{worst} (red), S^{worst} (green), U_S^{worst} (black).

5.5 Usability of anonymized speech data

From the users' perspective, the usability of anonymized speech is often measured in terms of the performance of models trained on it. These models must predict various attributes, such as verbal content, emotions,

etc., in the speech signal with similar accuracy as models trained on original speech. Section 5.3.4.2 presents one such use-case where the ASR models trained solely on anonymized speech are compared to the models trained on original speech. Although the "A-A" bars in Figure 5.12 show that the $\text{ASR}_{\text{eval}}^{\text{anon}}$ model decodes anonymized speech almost as accurately as the ASR_{eval} model decodes original speech, the "O-A" bars indicate that the $\text{ASR}_{\text{eval}}^{\text{anon}}$ model doesn't generalize well to original speech. In this section, we conduct experiments to find whether this generalization loss can be recovered by augmenting the anonymized corpus with a certain amount or publicly available original speech.

Experimental setup We use different proportions of the original *train-clean-360* corpus, as described in Table 5.4, to augment the anonymized version of this data set. The design choices selected for anonymizing this data set are: {PLDA distance, *dense* proximity, *random* gender-selection, and *speaker-level* assignment}. For every experiment, utterances composing $p_c \in \{0, 10, 20, 50, 100\}\%$ of the total duration of the data set, i.e., 360 hours are randomly sampled from the original data set. The remaining duration is covered by the utterances from the anonymized corpus. Five mixed speech corpora are created, corresponding to $p_c = 10\%$, $p_c = 20\%$, $p_c = 2 \times 10\%$ where 10% of clean data is randomly sampled and then duplicated to occupy 20% of the duration, $p_c = 50\%$, and $p_c = 5 \times 10\%$ where 10% of clean data is randomly sampled and then duplicated five times to occupy 50% of the duration. The ASR models trained over these five mixed corpora are compared with the models trained solely over original speech ($p_c = 100\%$) and anonymized speech ($p_c = 0\%$). They are compared and evaluated in terms of the WER obtained over original as well as anonymized versions of the *dev-clean* and *test-clean* data sets described in Table 3.1.

Results Figure 5.20 shows the performance of the seven different $\text{ASR}_{\text{eval}}^{\text{anon}}$ models. We are mostly concerned with the first row of this figure since it shows the performance over original speech. The most interesting observation is the setting with $p_c = 10\%$, which validates the hypothesis that just 10% of original data is enough for the model to generalize well to original speech. A huge relative WER improvement of 51% is observed over the model trained with $p_c = 0\%$ on both the development and test sets, although this model still performs 25% worse than the baseline WER relatively.

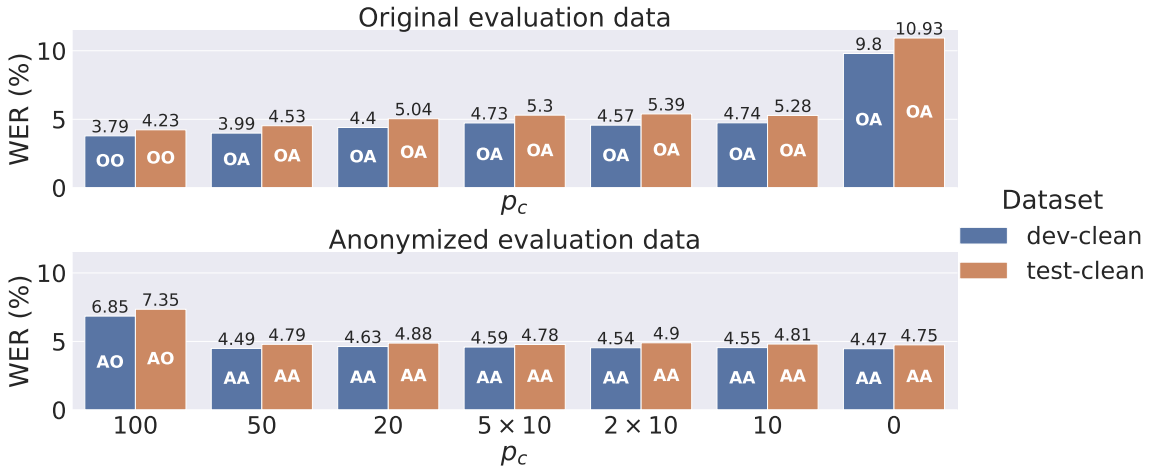


Fig. 5.20 Performance of the seven different $\text{ASR}_{\text{eval}}^{\text{anon}}$ models trained using different proportions of original data (p_c). The first and second rows indicate the performance of these models over the original and anonymized data, respectively. The text at the center of the bars specifies the decoding setting as described in Section 5.3.4.2.

Duplicating data ($p_c = 2 \times 10\%$ and $p_c = 5 \times 10\%$) does not translate into a significant performance gain, but increasing the proportion of unique data brings the performance of the models much closer to the baseline performance. The model trained with $p_c = 20\%$ performs only 16% (development) and 19% (test) worse relatively to the baseline WER, while the gap further reduces to 4% for development and 5% for test when $p_c = 50\%$.

5.6 Summary

We introduced four design choices for x-vector based speaker anonymization. The effect of each design choice was studied with respect to privacy and utility metrics of the anonymized speech and the optimal combination of choices was recommended. Experiments showed that the anonymized speech corpus is suitable to train a viable ASR model and that a reasonable amount of privacy protection is achieved even if a *Semi-Informed* attacker attempts to re-train the ASV model with an anonymized speech corpus.

We further investigated pitch conversion to remove the residual speaker information present in the pitch sequence of the anonymized speech and to enhance the naturalness of the synthesized voice. While the linkability of the output anonymized speech is significantly reduced as compared to no conversion, a noticeable rise in the WER is observed after pitch conversion. Nonetheless, this is a promising direction for future research which needs further exploration for a pitch conversion method that preserves privacy without degrading the utility. The newly proposed percentile-based pitch transformation method outperforms the conventional Gaussian normalization method and the newly explored minmax scaling in terms of privacy as well as utility.

We assessed the robustness of the proposed anonymization scheme as a function of the number of enrollment speakers. We conducted closed-set ASI by incrementally adding thousands of speakers in the population and observed that the rank of the true speaker quickly increases and converges close to chance-level performance after anonymization. Another interesting observation can be made using top-k membership analysis: the loss of precision before anonymization that is seen after adding thousands of speakers in the enrollment set is equivalent to adding only a couple of speakers after anonymization. Specifically, the best combination of design choices offers the same level of protection against re-identification attacks among 50 speakers as original speech among 20,000 speakers.

We performed the worst-case privacy protection assessment of the proposed anonymization scheme and conclude that a majority of utterances are well protected after anonymization, raising the lower bound of privacy significantly. Finally, we conducted a usability study which shows that ASR models trained over anonymized speech can generalize well to original speech if the anonymized training set is augmented with a small amount of original speech.

A fundamental assumption made in this chapter was that, out of the three features extracted from the speech signal, i.e., pitch, BN features, and x-vector, speaker-related information concentrates in the x-vector and replacing it with a new pseudo-speaker will delete most identity markers of the speaker in the synthesized speech. Our results on pitch conversion (Section 5.3.5) suggest that it is not the case. We further challenge this assumption in Chapter 6, and propose techniques to remove the residual speakers' identity from the pitch and BN features.

Chapter 6

Removing Residual Speaker Information — Towards Provable Guarantees

> There is nothing like looking, if you want to find something.
>
> *J.R.R. Tolkien*

In the previous chapter, we introduced the x-vector based anonymization scheme which was used as the primary baseline for the first VoicePrivacy challenge. Recall that this scheme seeks to separate the speaker identity information from the intonation and linguistic content so as to generate speech where only the identity information has been removed. It relies on the extraction of three types of features from a speech recording: (i) an x-vector which encodes the characteristics of the speaker's voice, (ii) a sequence of BN features that captures fine-grained linguistic information, and (iii) a sequence of pitch features which conveys the intonation [106]. Speaker anonymization is then realized by re-synthesizing speech from the BN and pitch features of the original speech recording and a replaced speaker embedding corresponding to another (real or pseudo) speaker. This approach is considered as the baseline anonymization scheme in this chapter. While this general approach has been quite successful and achieves good practical performance as seen in the previous chapter, the *disentanglement* of the speaker information is not perfect: intonation and linguistic features are known to contain residual identity information [79] which can propagate to the anonymized speech and be used by an adversary to re-identify speakers (see Section 6.2). Furthermore, the effectiveness of anonymization is evaluated only *empirically*: even if the evaluation is performed using state-of-the-art automatic speaker identification (ASI) or ASV techniques and takes into account some auxiliary information that the adversary may have (attacker's knowledge as defined in Section 3.1), there is no provable guarantee that the resulting speech cannot be de-anonymized using better attacks.

In this chapter, we remove speaker information from pitch and BN features by designing feature extractors that satisfy differential privacy (DP) using careful addition of noise in intermediate layers. We plug these extractors in the baseline anonymization pipeline and generate, for the first time, differentially private utterances with a provable upper bound on the speaker information they contain. Plugging our private feature extractors into the full speaker anonymization pipeline, we directly obtain a DP version of the x-vector based speaker anonymization scheme, whose speaker information is analytically bounded by the DP parameter ϵ.

We evaluate empirically the privacy and utility resulting from our DP anonymization scheme on the LibriSpeech data set. Experimental results show that the generated utterances are intelligible while protected

against strong attackers who have significant knowledge of the anonymization process. In order to interpret the effect of our analytical ϵ-DP guarantee [138, 136, 212], we conduct a two-step evaluation of practical privacy and utility. First, we evaluate empirically the amount of speaker information retained in pitch and BN features by training an ASI model directly on pitch and BN features. We compare the standard (non-private) features used in the previous chapter to the features output by our proposed DP feature extractors. Second, plugging our feature extractors into the best anonymization scheme proposed in Chapter 5, we show that we can generate speech utterances which empirically preserve better the privacy of speakers (in line with our analytical privacy guarantees) at only a small cost in utility. Practical privacy is measured by the equal error rate (EER) and linkability ($D_{\leftrightarrow}^{\text{sys}}$) achieved by a state-of-the-art ASV system, while utility is measured by the word error rate (WER) of an ASR system trained and tested on anonymized speech. Low WER/$D_{\leftrightarrow}^{\text{sys}}$ and high EER indicate that the speech generated with our DP approach can be shared, stored, annotated and used to train ASR models, while protecting the speaker identity in voice-based services.

In summary, the contributions made in this chapter advance the state-of-the-art in speaker anonymization as follows:

- We empirically show that the standard BN and pitch features contain a lot of speaker information, obtaining 97% and 37% recognition accuracy on a data set of 921 speakers when training ASI models directly on these features. We analytically introduce DP-pitch and DP-BN feature extractors that remove speaker information with analytical privacy guarantees and preserve the intonation and linguistic information of BN and pitch features. For instance, our DP extractors with a privacy guarantee of $\epsilon = 1$ reduce the accuracy of ASI models on BN and pitch to 14% and 5%, respectively.

- We show that our DP-BN features can be shared instead of raw utterances for both training and inference of ASR models with reasonable effect on the WER. On LibriSpeech, our DP-BN extractor with $\epsilon = 1$ achieves 6% WER, compared to the 4% WER of the original BN features.

- We synthesize speech with an analytical privacy guarantee from our DP-pitch, DP-BN and replaced x-vector using the speaker anonymization pipeline proposed in Chapter 5, studying the effect of each component. We show that the privacy is protected even against a strong *Semi-Informed* attacker, while utility remains high (and can even slightly improve). Using the LibriSpeech data set, our DP speech can achieve an EER and a WER of 48.9 and 5.6%, respectively, whereas the anonymization scheme without the DP components achieves an EER and a WER of respectively 47.1 and 6.8%, without any provable privacy guarantee.

This chapter is organized as follows. In Section 6.1, we provide an overview of our method, followed by the detailed presentation of our DP-pitch and DP-BN extractors. In Section 6.2, we evaluate the privacy and utility of our approach compared to the baseline speaker anonymization scheme. We summarize the findings in Section 6.3.

The techniques proposed in this chapter are a collaborative effort, where the approach for DP-pitch was formulated by Dr. Ali Shahin Shamsabadi[1], and the DP-BN approach was essentially contributed by the author of this thesis.

6.1 Proposed approach

In this chapter, we propose to use ideas from differential privacy (DP) [69], a rigorous mathematical framework to quantify the information leakage of algorithms, to design more robust speaker anonymization

[1] https://alishahin.github.io/

schemes. Refer to Section 2.4 for a detailed account of DP. Following the pipeline of baseline anonymization scheme described above, we introduce DP-pitch and DP-BN feature extractors that can bound the speaker identity of the intonation and linguistic attributes used to re-synthetize speech. Instead of adding noise directly to the original features (which would destroy the linguistic and intonation content that we wish to preserve), we propose to train machine learning models that extract such features in a DP fashion while maximizing their utility. For pitch, we introduce an autoencoder network that includes a Laplace noise layer between the encoder and the decoder and learns to reconstruct its input pitch at the output. We use a new reconstruction loss function that maximizes the correlation between the input and reconstructed pitch, so as to preserve the global pitch dynamics which conveys intonation (e.g., pitch increases when asking a question) while perturbing its local variations which are more specific to each speaker [3, 226, 54, 194]. Regarding the BN features, we train an ASR acoustic model with a Laplace noise layer placed after the intermediate layer that corresponds to the BN features. In this way, our DP-BN features are trained to retain as much as possible the phonetic information needed to decode the linguistic content while the noise helps to remove the residual speaker information. In addition to DP-pitch and DP-BN, we choose a public x-vector randomly and independently of the input utterance so that it does not contain any information about the original speaker, following a modified version of the 'dense' strategy proposed in Section 5.3.2.2.[2]

6.1.1 Overview

Our speaker anonymization approach is depicted in Figure 6.1. DP-pitch and DP-BN features are first extracted from the input speech. These features, along with a target speaker embedding (x-vector) that corresponds to a different (pseudo) speaker, are then used to re-synthesize speech using acoustic and neural source filter (NSF) synthesis models. Note that the x-vector is chosen independently of the input utterance.[3] Therefore, information about the input speaker can only leak through pitch and BN features. Our contribution is to design pitch and BN feature extractors that satisfy DP so as to provably upper bound the amount of residual speaker information embedded in these features while preserving intonation and linguistic content. By the post-processing property of DP, we guarantee that our end-to-end pipeline (from the input speech to the anonymized speech) also satisfies DP.

Our DP-pitch extractor consists of a conventional pitch estimator (here, YAAPT, as explained in Section 2.2.1) followed by an autoencoder network with a Laplace noise layer trained to reconstruct the global pitch dynamics using a customized loss function. Our DP-BN extractor is an ASR acoustic model, also with a Laplace noise layer, trained on speech utterances to estimate the corresponding word sequence. We use a public set of annotated speech utterances to train both extractors prior to deployment.

We emphasize that our extractors are quite generic. They may be used in variants of the same speaker anonymization pipeline [196, 37, 300, 75]. They can also be used independently: for instance, our DP-BN features are sufficient to decode the linguistic content at inference time, as we will show in our experiments.

In the rest of this section, we introduce some useful notations, and describe our DP-pitch and BN feature extractors in detail.

Notations Recall that we assumed $\mathbf{s}$ to be a speech utterance comprising T time frames. The value of T depends on the duration of the utterance and the chosen frame rate (typically, 10 ms). The pitch sequence

[2] To select the pseudo-speaker based on the distance from the source speaker violates DP, since the source information is being used to generate pseudo-speaker. Similarly, for same or opposite gender-selection, we are bound to use the source information. This selection may also have an impact on the utility, hence *dense* proximity with *random* gender-selection is preferred, but the closest cluster to source is not filtered out.

[3] This scheme, i.e., dense proximity, random gender-selection and speaker-level assignment was shown to give best performance in Chapter 5. Alternatively, one could select a target x-vector close to the x-vector of the original speaker in a differentially private way, as proposed in [110]. However, we observed no clear gain in utility.

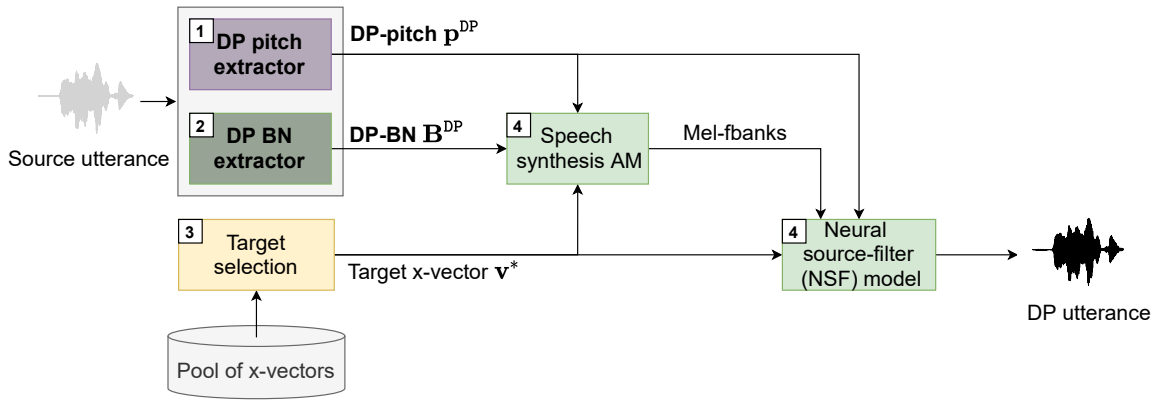


Fig. 6.1 Overview of our proposed DP speaker anonymization scheme. Our main contributions are the DP-pitch and DP-BN feature extractors (shown as blocks 1 and 2), which make the full pipeline DP.

computed from $\mathbf{s}$ is a non-negative 1-dimensional sequence of length T, which we denote by a vector $\mathbf{p} \in \mathbb{R}_+^T$. The BN features extracted from $\mathbf{s}$ are an M-dimensional sequence of length T that we denote by a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_T]^\top \in \mathbb{R}^{T \times M}$ where each $\mathbf{b}_t \in \mathbb{R}^M$. Throughout this section, we assume that we have access to a *public* data set $\mathcal{X} = \{(\mathbf{s}_i, W_i)\}_{i=1}^N$ of N annotated speech utterances to train our DP feature extractors. For a given utterance index $i \in \{1, \ldots, N\}$, $\mathbf{s}_i$ denotes the speech waveform and W_i denotes the corresponding ground truth text transcription.

6.1.2 Differentially-private pitch extractor

As mentioned earlier, the global dynamics of the pitch sequence $\mathbf{p}$ for an utterance $\mathbf{s}$ conveys intonation, while its local variations are more specific to each speaker [3, 226, 54, 194]. We aim to learn a DP autoencoder $\mathcal{A}$ which takes as input a raw pitch sequence $\mathbf{p}$ computed by a conventional pitch estimator and outputs a perturbed pitch sequence $\mathbf{p}^{\text{DP}}$ of the same length in which the identity information has been removed while most of the intonation is preserved. An obvious approach to obtain a DP autoencoder is to rely on output perturbation, i.e., to add Laplace noise directly to the raw pitch $\mathbf{p}$. However, aside from the difficulty of bounding the ℓ_1-sensitivity of pitch sequences in a tight manner, this baseline strategy would largely destroy the time correlations that are indicative of intonation elements that we wish to preserve.

Instead, we propose to use a deep convolutional autoencoder with a noise layer. Below, we describe the architecture of our autoencoder, explain how it is trained, and finally how it can be deployed to anonymize pitch sequences. The block diagram of our proposed DP-pitch extractor is shown in Figure 6.2.

Autoencoder architecture We propose to define $\mathcal{A} = \mathcal{D} \circ \mathcal{N}_p \circ \mathcal{E}$ as a fully convolutional autoencoder composed of an encoder $\mathcal{E}$, a noise layer $\mathcal{N}_p$ and a decoder $\mathcal{D}$. This architecture is inspired by [257]. The benefit of using only convolutional layers is two-fold. First, a fully convolutional architecture enables us to deal with variable-length input and output sequences as the shape and size of the weights of each convolutional layer (kernel) are not affected by the size of the input and output of that layer. Second, convolutional layers are suitable to capture time dependencies in pitch sequences.

The encoder $\mathcal{E}$ maps an input pitch sequence $\mathbf{p} \in \mathbb{R}^T$ to a latent representation $\mathbf{h} \in [0, 1]^{C \times T}$:

$$\mathbf{h} = \mathcal{E}(\mathbf{p}) \tag{6.1}$$

through 3 convolutional layers (each with C channels) with sigmoid activation functions.

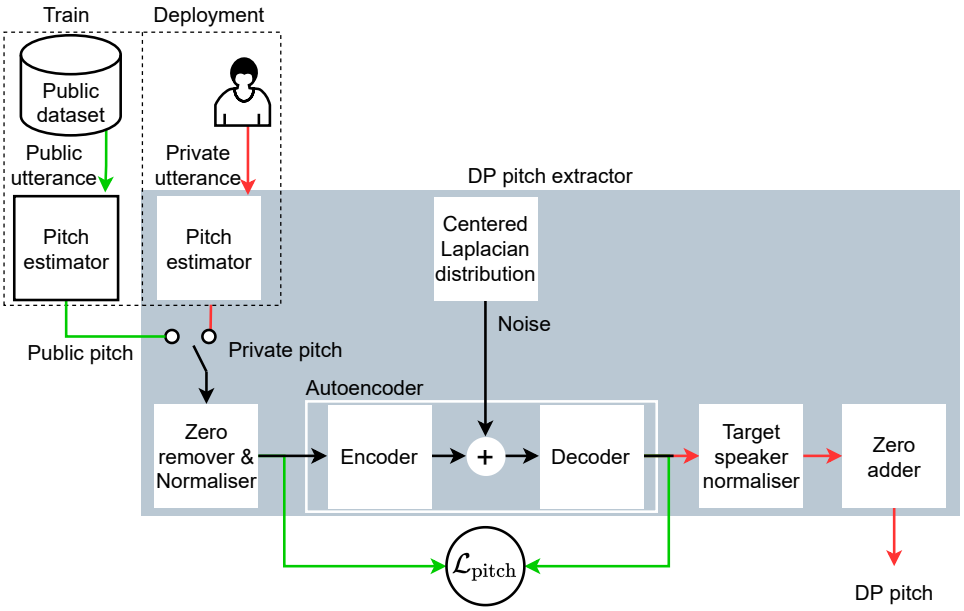


Fig. 6.2 Proposed DP-pitch extractor. The convolutional autoencoder with a noise layer is trained using public pitch sequences and subsequently used to generate perturbed pitch sequences from private pitch sequences in a differentially private fashion. Black arrows show paths that are common to both training and deployment, while green and red arrows apply only to training or deployment, respectively.

In order for the autoencoder $\mathcal{A}$ to satisfy ϵ-local DP for a given $\epsilon > 0$, the encoder is followed by a noise layer $\mathcal{N}_p$ which adds centered Laplace noise to each entry of the latent representation $\mathbf{h}$ to generate a perturbed latent representation $\mathbf{h}^{\text{DP}} \in \mathbb{R}^{C \times T}$:

$$\mathbf{h}^{\text{DP}} = \mathcal{N}_p(\mathbf{h}) = \mathbf{h} + \text{Lap}(\Delta_1(\mathcal{E})/\epsilon), \tag{6.2}$$

where $\Delta_1(\mathcal{E}) = \max_{\mathbf{p},\mathbf{p}'} \|\mathcal{E}(\mathbf{p}) - \mathcal{E}(\mathbf{p}')\|$ is the ℓ_1-sensitivity of $\mathcal{E}$. While tightly bounding the sensitivity of neural networks can be challenging in general [221], here the use of the sigmoid activation allows us to easily bound $\Delta_1(\mathcal{E})$ since each entry of $\mathbf{h}$ belongs to $[0, 1]$:

$$\Delta_1(\mathcal{E}) = C \times T \times 1 = CK. \tag{6.3}$$

This bound is tight enough in practice for the Laplace noise injected to the features not to be detrimental to utility, as long as the value of ϵ remains reasonable (away from zero).

Finally, the decoder $\mathcal{D}$ takes as input the perturbed latent representation $\mathbf{h}^{\text{DP}}$, deterministically clips each of its entries back to $[0, 1]$ (which we found to help training to converge), and decodes it into a perturbed pitch sequence $\mathbf{p}^{\text{DP}} = \mathcal{D}(\mathbf{h}^{\text{DP}})$ through 3 convolutional layers (two C-channel layers with sigmoid activation followed by one layer with linear activation).

Training phase We train our autoencoder on a set of raw pitch sequences $\{\mathbf{p}_i \in \mathbb{R}^{T_i}\}_{i=1}^N$ computed from the speech waveforms $\mathbf{s}_i$ in the public data set $\mathcal{X}$ using for instance the YAAPT estimator. The pitch sequences are pre-processed as follows. First, zero values are removed. Indeed, these values indicate silence or unvoiced phonemes and must be kept fixed to zero so that silence remains silence, and every unvoiced

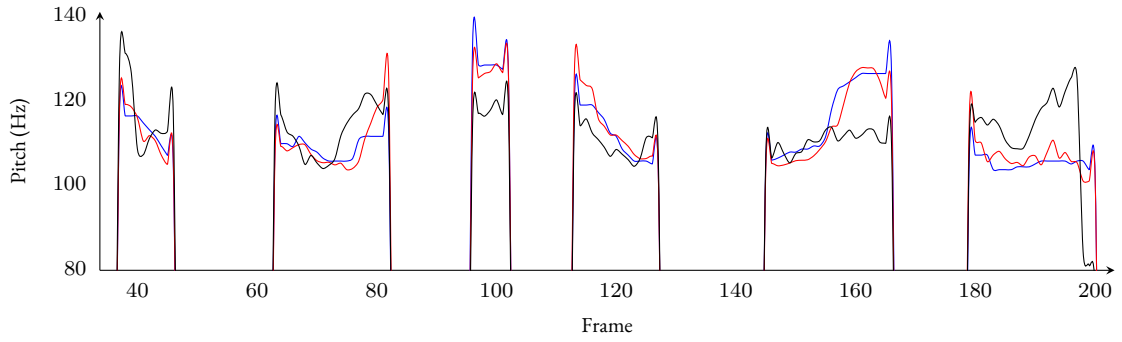


Fig. 6.3 Visualization of the original (non-private) pitch sequence (——) and noisy reconstructed pitch sequences obtained with our approach for $\epsilon = 10$ (——) and $\epsilon = 1$ (——). In general, our approach preserves the dynamics of the original pitch sequence thanks to our correlation-based loss function.

phoneme remains the same unvoiced phoneme.[4] This operation was also performed and explained in Section 5.14. It also makes it possible to account for variation of pitch across successive voiced phonemes. Table 6.1 shows statistics on the length of pitch sequences and the proportion of zero values in a data set used in our evaluation, and Figure 6.3 shows an example of raw pitch sequence. After removing zeros, since pitch differs in range across speakers, each sequence is normalized to zero mean and unit variance.

Table 6.1 YAAPT pitch statistics on the dev-clean subset of LibriSpeech [220].

	Min	**Max**	**Avg**	**Std**
Length T_i	147	3261	743	493
Non-Zeros	24%	76%	53%	8%

To preserve the intonation in the reconstructed pitch, we propose to train the autoencoder by minimizing the following loss function:

$$\mathcal{L}_{\text{pitch}} = 1 - \sum_{i=1}^{N} \texttt{Corr}(\mathbf{p}_i, \mathbf{p}_i^{\text{DP}}), \tag{6.4}$$

where $\texttt{Corr}(\cdot,\cdot)$ is the Pearson correlation coefficient. The justification for this choice of loss is that maximizing correlations between the original and reconstructed noisy pitch will make reconstruction errors in the local variations of the pitch (which can be speaker-specific) less costly than in the global dynamics of the sequence (which convey intonation). We refer to Figure 6.3 for an illustration.

Deployment phase For any private utterance $\mathbf{s}$, similarly to the training phase, we compute the pitch sequence $\mathbf{p}$, remove the zeros, normalize it, and push it to the autoencoder to obtain a perturbed pitch sequence $\mathbf{p}^{\text{DP}}$. We normalize the perturbed sequence using Equation (3.3) to match the mean μ_{tgt} and variance σ_{tgt} of the pitch of a target (pseudo) speaker, where μ_{tgt} and σ_{tgt} are computed over a public set of utterances from the target speaker. In the context of the full speaker anonymization pipeline of Figure 6.1, this normalization (which we call *pitch conversion*) makes the mean and variance of the perturbed pitch consistent with the choice of the target x-vector. Finally, we add the zero values back in their original positions in the sequence.

[4] For instance, replacing the zero pitch on phoneme /p/ by a nonzero pitch would transform it into a /b/.

Privacy guarantees By the Laplace mechanism (Definition 3), $\mathcal{N}_p \circ \mathcal{E}$ satisfies ϵ-DP, and so does the autoencoder $\mathcal{A} = \mathcal{D} \circ \mathcal{N}_p \circ \mathcal{E}$ by the post-processing property of DP.

6.1.3 Differentially-private BN extractor

We now turn to the BN features, which are phonetic features that should be sufficient to decode the linguistic information. BN features are typically obtained as an intermediate layer of an ASR acoustic model. However, traditional BN features also contain residual speaker information. We propose to address this issue by adding a noise layer, similarly to the approach used for pitch.

ASR model architecture We adapt the widely used ASR acoustic model architecture and sequence-discriminative training criterion described as the "Conventional approach" in Section 2.2.3. For clarity, we split the ASR acoustic model $\mathcal{M} = \mathcal{T} \circ \mathcal{N}_B \circ \mathcal{B}$ into three sequential parts: a BN extractor $\mathcal{B}$, followed by a noise layer $\mathcal{N}_B$, and finally a triphone classifier $\mathcal{T}$ (see Figure 6.4). The BN extractor $\mathcal{B}$ takes as input a sequence of acoustic features $\mathbf{O} \in \mathbb{R}^{T \times A}$ extracted from a speech utterance $\mathbf{s}$ with T frames and outputs a sequence of BN features $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_T]^\top \in \mathbb{R}^{T \times M}$:

$$\mathbf{B} = \mathcal{B}(\mathbf{O}). \tag{6.5}$$

The acoustic features $\mathbf{O}$ are the concatenation of 40-dimensional MFCCs and 100-dimensional i-vectors [251] which help the acoustic model adapt to different speakers. Therefore, the per-frame dimensionality of these features is $A = 140$. The BN extractor $\mathcal{B}$ is composed of 17 TDNN-F layers, which perform one-dimensional convolution operations to learn the temporal context present in the acoustic feature sequence $\mathbf{O}$.

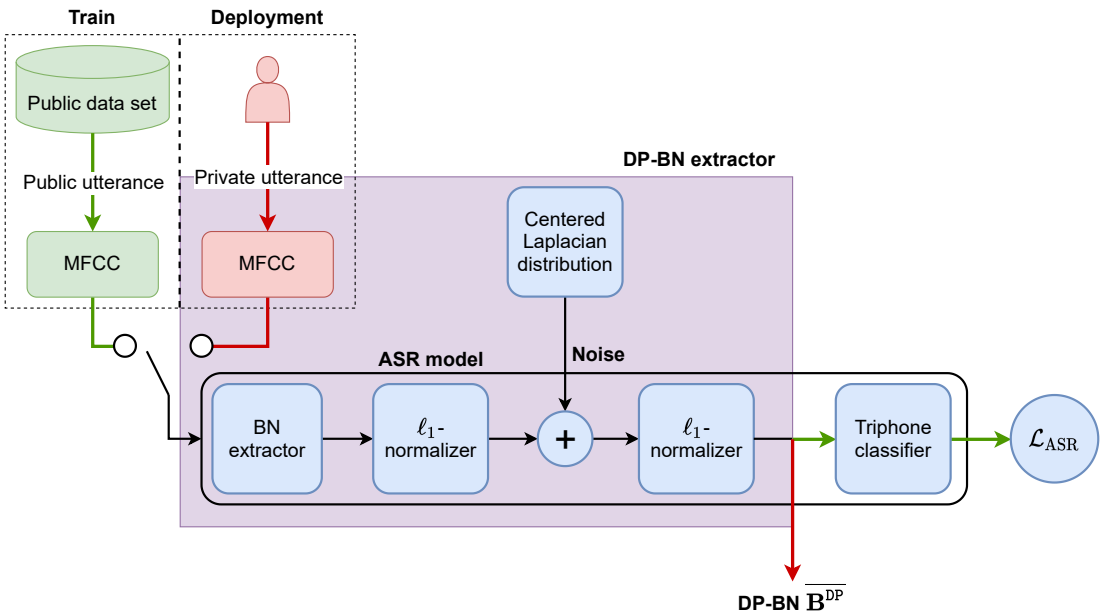


Fig. 6.4 Proposed DP-BN extractor. The ASR acoustic model with a noise layer is trained on public utterances and subsequently used to generate perturbed BN features from private utterances in a DP fashion. Black arrows show paths that are common to both training and deployment, while green and red arrows apply only to training or deployment, respectively.

We now describe our noise layer $\mathcal{N}_B$, which we use to hide speaker information and achieve DP. Each frame-level BN feature vector $\mathbf{b}_t$ is M-dimensional with $M = 256$. Due to this high dimensionality, we propose to enforce ϵ-DP at the frame level rather than at the utterance level (we will experimentally show that frame-level DP is sufficient). Note also that each frame-level BN feature vector $\mathbf{b}_t$ is not normalized. Therefore, our noise layer $\mathcal{N}_B$ first normalizes each $\mathbf{b}_t$ to have unit ℓ_1-norm and then adds Laplace noise to their entries to generate a sequence of perturbed BN features $\mathbf{B}^{\text{DP}} = [\mathbf{b}_1^{\text{DP}}, \ldots, \mathbf{b}_T^{\text{DP}}]^\top \in \mathbb{R}^{T \times M}$:

$$\mathbf{B}^{\text{DP}} = \mathcal{N}_B(\mathbf{B}) = \begin{bmatrix} \mathcal{N}_b(\mathbf{b}_1) \\ \vdots \\ \mathcal{N}_b(\mathbf{b}_T) \end{bmatrix}, \tag{6.6}$$

where $\mathcal{N}_B(\mathbf{b}) = \frac{\mathbf{b}}{\|\mathbf{b}\|_1} + \text{Lap}(2/\epsilon)$. The scale of the centered Laplace noise comes from the fact that the ℓ_1-sensitivity of the normalized frame-level BN features is bounded by 2. Thereafter, $\mathbf{B}^{\text{DP}}$ is further ℓ_1-normalized to obtain $\overline{\mathbf{B}^{\text{DP}}}$.

Finally, the triphone classifier $\mathcal{T}$ takes the sequence of perturbed BN features $\overline{\mathbf{B}^{\text{DP}}}$ as input and outputs the corresponding triphone log-posterior probabilities $\{P(S_k|\overline{\mathbf{B}^{\text{DP}}})\}_{k=1}^N$ which represent the phonetic content of $\mathbf{s}$. We refer to Section 2.2.3 for details on the architecture of $\mathcal{T}$.

Training phase Our ASR acoustic model $\mathcal{M}$ is trained on acoustic features $\{\mathbf{O}_i\}_{i=1}^N$ extracted from the utterances $\{\mathbf{s}_i\}_{i=1}^N$ in the public annotated data set $\mathcal{X}$ to output the corresponding transcriptions $\{W_i\}_{i=1}^N$. To preserve the linguistic content, we minimize a cost function $\mathcal{L}_{\text{ASR}} = \mathcal{L}_{\text{MMI}} + 0.1 \cdot \mathcal{L}_{\text{CE}}$ composed of two terms defined in Equations (2.17) and (2.18). The only difference for training with the noise layer is that the term $P(\mathbf{O}_i|S_i)$ in the numerator and denominator of $\mathcal{L}_{\text{MMI}}$ is now conditioned upon the DP-BN features $\overline{\mathbf{B}^{\text{DP}}}$, i.e., $P(\mathbf{O}_i|S_i) \propto \prod_{t=1}^{T_i} P(S_{i,t}|\overline{\mathbf{B}_i^{\text{DP}}})/P(S_{i,t})$. Similarly, the computation of the $\mathcal{L}_{\text{CE}}$ term is modified as:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N}\sum_{t=1}^{T} \log P(S_{i,t}|\overline{\mathbf{B}_i^{\text{DP}}}). \tag{6.7}$$

As mentioned previously, the noise layer $\mathcal{N}_B$ normalizes the feature vectors $\mathbf{b}_t$ to have unit ℓ_1-norm before and after the application of noise. The gradients backpropagating from this layer are derived in Appendix B.1.

Deployment phase For any private utterance $\mathbf{x}$ with acoustic features $\mathbf{O}$, we can use the trained ASR model to generate a sequence of perturbed BN features $\overline{\mathbf{B}^{\text{DP}}} = \mathcal{N}_B \circ \mathcal{B}(\mathbf{O})$.

Privacy guarantees By the Laplace mechanism, the frame-level mechanism $\mathcal{N}_B$ satisfies ϵ-DP. Note that this frame-level guarantee can be converted into an utterance-level guarantee using the composition property of DP: the BN extractor $\mathcal{N}_B \circ \mathcal{B}$ satisfies $T\epsilon$-DP for utterances of length T.

6.2 Empirical validation

Our DP-pitch and BN extractors aim to remove the residual identity information from the intonation and linguistic attributes of an utterance, which are then used in our DP speaker anonymization pipeline of Figure 6.1 to output utterances with rich intonation and linguistic attributes. Therefore, our experiments consider the following major dimensions:

1. How much identity information is retained within the original pitch, BN features and anonymized utterances?
2. How well does DP bound the information about the identity within pitch, BN features and utterances?
3. How does DP affect the utility of utterances and BN features for training and deployment of ASR models?

6.2.1 Experimental setup

Data set Following the framework introduced in the previous chapter, we use different subsets of the LibriSpeech corpus (Ref. Table 3.1) for training our DP extractors, the attack and evaluation models. The details of subsets used to train different modules of the framework are mentioned in Table 5.6. We mention the data sets to train the new DP extractors in the next paragraph.

Implementation details All feature sequences have a frame rate of 10 ms. For pitch estimation, we use YAAPT[5] as explained in Section 2.2. We implement our proposed DP-pitch autoencoder in PyTorch and train it on *train-clean-100*, using a mini-batch size of 1 due to the variable sequence length. We use the Adam optimizer [153] with a learning rate of 10^{-3}, a weight decay of 10^{-5}, and a dropout value of 10^{-3}, similarly to [257]. We implement our proposed DP-BN extractor using the Kaldi toolkit [230] and train it on the combination of *train-clean-100* and *train-clean-500* as the ASR model in Table 5.6.

Privacy metrics In addition to providing an analytically provable privacy guarantee in the form of ϵ-DP, we empirically measure privacy to interpret the value of ϵ [138, 136, 212] in two different ways.

First, the practical privacy achieved by the DP-pitch and BN extractors is measured by the classification accuracy (ACC) of a closed-set ASI system, i.e., the proportion of utterances which are assigned to the correct speaker, which varies between 0% (best) and 100% (worst). The ASI system follows the classical TDNN speaker classification architecture[6] in Kaldi [269] except that, instead of MFCCs as inputs, it is trained on pitch or BN features, before and after applying DP, extracted from the *train-clean-360* data set. This data set contains 921 speakers and it is divided into train/valid/test splits such that 80% utterances of each speaker are used for training, 10% for validation and 10% for testing. In the case of pitch, silent regions are removed using energy-based voice activity detection before training the ASI. We train the ASI system over the training split with 15 epochs using a mini-batch size of 64. The validation set is used for monitoring the generalization performance and early stopping in case of convergence. We report the ACC metric over the test split.

Second, the practical privacy achieved by the whole DP anonymization pipeline is measured by the EER and $D_{\leftrightarrow}^{\text{sys}}$ achieved by an ASV system. These metrics are explained in detail in Section 3.4.1. We recall (see Section 2.2.5 for details) that ASV computes the PLDA log-likelihood ratio score given the x-vectors of trial and enrollment utterances, and decides whether they are from the same speaker by comparing it with a threshold. The EER is equal to the false acceptance rate and the false rejection rate at the threshold for which these two rates are equal [26], and it varies between 0% (worst) and 50% (best). The linkability measures the amount of overlap between the distributions of same-speaker (mated) and different-speaker (non-mated) scores. It varies between 0 (best) and 1 (worst). The ASV system, i.e., both the x-vector extractor (a TDNN with MFCCs as inputs) and PLDA, follows the classical ASV setup in Kaldi [230]. It was

[5] `http://bjbschmitt.github.io/AMFM_decompy/pYAAPT.html`

[6] The network architecture is presented in Table 2.1.

previously established in Chapter 3 that the speaker's identity can be revealed after anonymization if the attacker has knowledge about the anonymization scheme (*Semi-Informed* setting). Indeed, it allows the attacker to generate a large set of anonymized utterances to train a speaker recognition system which tries to discriminate between speakers even after anonymization. In order to evaluate privacy in this challenging context, we train the ASV system (both x-vector extractor and PLDA) over the *train-clean-360* data set anonymized using either DP-pitch or DP-BN features. We anonymize the training set using exactly the same method and parameters as used for the evaluation set. In total, we train 8 different ASV systems for the different values of ϵ (4 for DP-pitch and 4 for DP-BN features). We compute PLDA scores (Eq. (2.26)) using x-vectors extracted with these models and report EER and $D_{\leftrightarrow}^{\text{sys}}$.

Utility measures We quantify the preservation of linguistic content in utterances by the word error rate (WER) of an ASR system, i.e., the percentage of word substitutions, deletions, and insertions compared to the number of words in the ground truth transcriptions.

To train the evaluation ASR, we adapt the Kaldi recipe used for training an ASR system over the LibriSpeech data set. The training procedure and architecture of the ASR system is similar to the one described in Section 6.1.3, except that we do not use any noise layer after the BN extractor: the BN features are directly fed to the triphone classifier to compute the loss $\mathcal{L}_{\text{ASR}}$. We train 4 different ASR systems over the *train-clean-360* data set anonymized using DP-BN features with different values of ϵ. The TDNN acoustic model of each ASR system was trained for 4 epochs. We do not re-train the ASR system specifically for DP-pitch as we observed that the effect on the utility of the ASR system is minor. During decoding, we use a large trigram language model $P(W)$ available at the openslr website.[7]

Methods under comparison We compare the practical privacy and utility of our proposed DP speaker anonymization approach against the anonymized utterances output by the anonymization scheme[8] introduced in Chapter 5, which is referred to as the "baseline" in the following. Recall that this baseline anonymization scheme follows the same pipeline as in Figure 6.1 but uses regular (non-DP) pitch and BN extractors.

Table 6.2 Different instantiations of anonymization scheme and our DP extractors. Anon: Anonymized; PC: Pitch Conversion; DP: Differential Privacy.

Name	**Baseline** (Chapter 5)	**DP-pitch extractor**	**PC**	**DP-BN extractor**
`Anon`	✓	-	-	-
`Anon+PC`	✓	-	✓	-
`Anon+DP-BN`	✓	-	-	✓
`Anon+DP-Pitch`	✓	✓	✓	-
`Anon+DP`	✓	✓	✓	✓

We consider different instantiations of our method and the baseline anonymization scheme (see Table 6.2). To analyze the impact of each DP extractor separately and all of them together, we use, i) `Anon+DP-BN`, a modification of the baseline anonymization scheme where the BN extractor is replaced by our DP-BN extractor; ii) `Anon+DP-Pitch`, a modification of the baseline anonymization scheme where the pitch extractor

[7] `http://openslr.org/11/`

[8] The design choices selected for the baseline scheme are {*PLDA* distance, *dense* proximity, *random* gender-selection, and *speaker-level* assignment}.

is replaced by our DP-pitch extractor followed by pitch conversion; and iii) `Anon+DP`, a modification of the baseline anonymization scheme where both the BN and pitch extractors are replaced by our DP-BN and DP-pitch extractors followed by pitch conversion to generate DP utterances. Note that we consider the original baseline anonymization scheme `Anon` (without pitch conversion) as the baseline for `Anon+DP-BN`. However, for comparison with `Anon+DP-Pitch`, we consider the baseline anonymization scheme with pitch conversion `Anon+PC`, to be fair in terms of pitch conversion.

6.2.2 Results and discussion

Privacy and utility of bottleneck features and pitch Figure 6.5 shows the practical privacy of BN features and pitch in terms of the classification accuracy (ACC) of an ASI system trained directly on them. The results clearly demonstrate that both original BN features and original pitch retain a lot of speaker information. BN features contain more identity information than pitch: for example, ASI can recognize the identity of speakers from original BN features with 95% accuracy (recall that there are 900+ different speakers). In contrast, the ACC of original pitch is 38%. The practical privacy of BN features and pitch improves significantly when using our DP approach. For instance, DP-BN features with an analytical privacy budget of $\epsilon = 1$ reduce ACC to around 14%. Figure 6.5 also confirms that a better analytical privacy bound results in better practical privacy. For example, the ACC of DP-BN features with $\epsilon = 100$ is 40%, and it decreases to 20% for $\epsilon = 10$. In order to study the impact of DP noise on the utility of BN features, Figure 6.6 shows the utility achieved by the ASR model trained over *train-clean-360* using original BN features ($\epsilon = \infty$) and DP-BN features ($\epsilon \in \{1, 10, 100\}$).

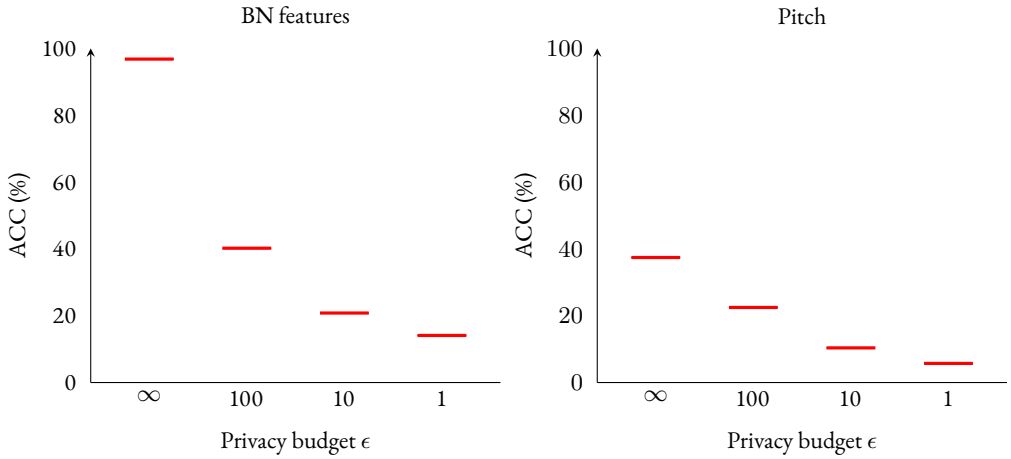


Fig. 6.5 Practical privacy evaluation of original BN features (∞) versus our proposed DP-BN features (left) and original pitch versus our proposed DP-pitch (right) for different privacy budgets of $\epsilon = 1$, $\epsilon = 10$ and $\epsilon = 100$. Practical privacy is assessed in terms of the classification accuracy of an ASI system (ACC) trained directly on BN features (left) and pitch (right) of the 921 speakers in the *train-clean-360* data set.

Strikingly, our DP-BN features have small effect on the WER of the ASR model ($\leq$ 2% absolute increase). Therefore, our DP-BN extractor outputs high-utility DP-BN features that contain enough linguistic information to perform the transcription task, while bounding speaker information as shown in Figure 6.5. We also show the correlation between DP-pitch and original pitch in Figure 6.6. The more noise injected in the pitch, the less correlation with the original pitch. We hypothesize that intonation is still preserved even with low correlation, which is confirmed by the negligible effect of DP-pitch on the ASR performance and when listening to anonymized utterances. Nonetheless, further experimentation

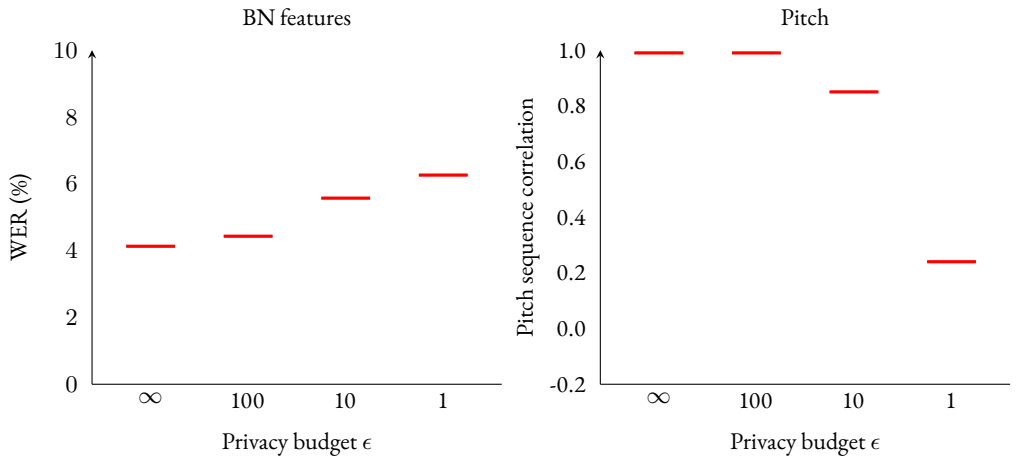


Fig. 6.6 Utility evaluation of original BN features (∞) versus our proposed DP-BN features (left) and original pitch versus our proposed DP-pitch (right) for different privacy budgets of $\epsilon = 1$, $\epsilon = 10$ and $\epsilon = 100$. The utility of BN features is assessed by the WER of an ASR system trained using the corresponding BN features. The utility of pitch is assessed by its correlation to the original pitch sequence.

is required to better quantify the amount of preserved intonation, for instance by training a network to classify certain prosodic attributes from pitch.

Privacy and utility of utterances Next, we evaluate the benefit of our proposed DP-pitch and DP-BN extractors when they are plugged in the baseline anonymization scheme. Figure 6.7 compares the practical privacy and utility of `Anon+DP-BN` for different ϵ against `Anon`. Recall that the ACC of `Anon+DP-BN` is lower than `Anon`. Similarly, decreasing ϵ increases the EER and decreases the $D_{\leftrightarrow}^{\text{sys}}$ of `Anon+DP-BN`. The WER of `Anon` is 4.7%. However, noise introduced by DP to the BN features increases the WER of `Anon+DP-BN` to 5.6%, 6.1% and 6.4% for privacy budgets of 100, 10 and 1, respectively.

Figure 6.8 shows the practical privacy and utility of `Anon+DP-Pitch` for different ϵ against `Anon+PC`. Decreasing ϵ in the DP-pitch extractor slightly improves the practical privacy of `Anon+DP-Pitch` in comparison to `Anon+PC`. The utility of `Anon+DP-Pitch` is similar to `Anon+PC`. Although the EER and the WER show a steady, monotonic rise as we decrease the ϵ indicating the privacy-utility trade-off, an unexpected drop is observed both in the value of $D_{\leftrightarrow}^{\text{sys}}$ and the WER for $\epsilon = 0.1$. The sudden drop indicates better privacy protection as well as better utility which is welcoming, but the explanation for this drop is not straightforward and needs further investigation because in our approach the DP noise is added to the intermediate layer of the autoencoder network, and not directly to the pitch values.

Finally, Table 6.3 compares the practical privacy and utility of `Anon+DP` utterances with the baseline `Anon+PC` anonymization scheme. The practical privacy of `Anon+PC` in terms of EER and $D_{\leftrightarrow}^{\text{sys}}$ is 47% and 0.14. All variants of `Anon+DP` increase the EER to close to its maximum possible value (50%) and achieve lower $D_{\leftrightarrow}^{\text{sys}}$ in the order of 0.10 thanks to our DP-pitch and DP-BN extractors that analytically bound the speaker information in the utterances. In addition to this, our `Anon+DP` yields highly intelligible speech: for some parameters, our system even achieves a WER of 5.6%, that is slightly better than `Anon+PC`.

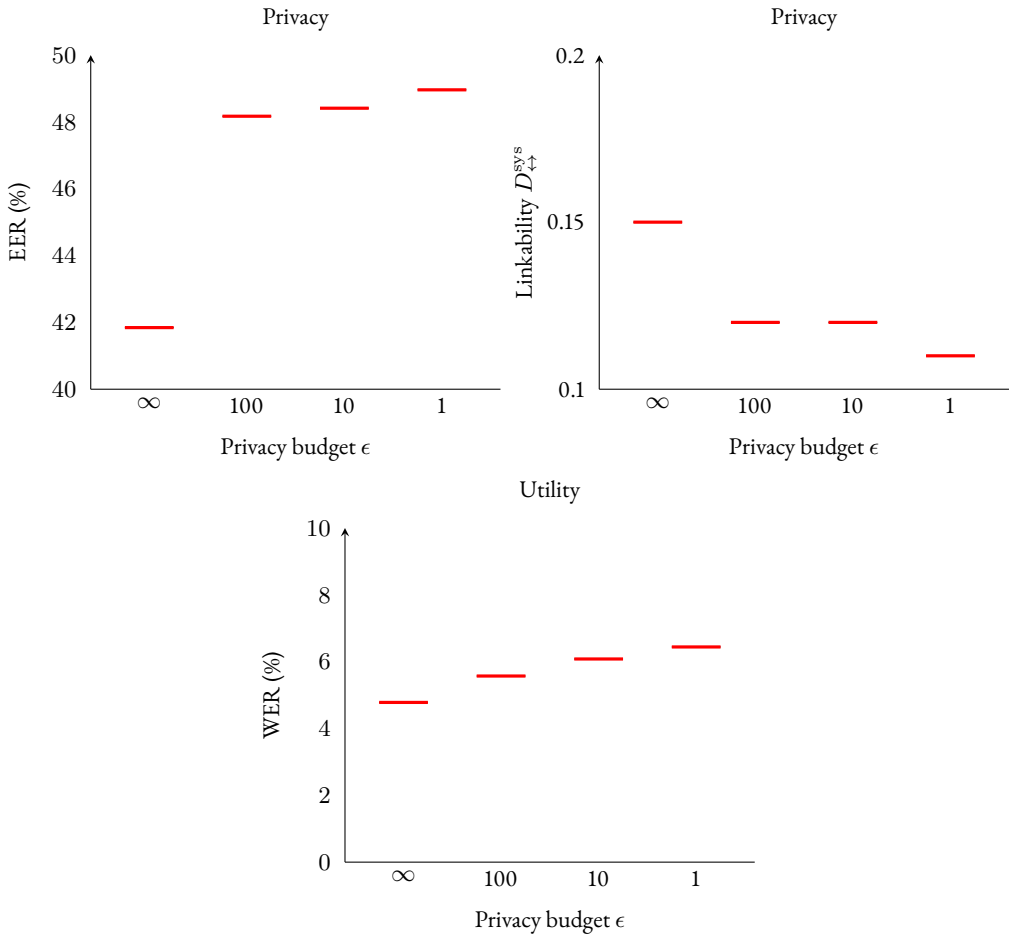


Fig. 6.7 Practical privacy in terms of EER and $D_{\leftrightarrow}^{\text{sys}}$, and utility in terms of WER of `Anon` and `Anon+DP-BN` for different ϵ of 100, 10 and 1. The ASR_{eval} system for each ϵ is trained on the anonymized utterances synthesized using the BN features of the same ϵ.

6.3 Summary

In this chapter, we proposed a DP speaker anonymization approach which can be used to share speech utterances for training or deployment of voice-based services while concealing the speaker's identity. More specifically, we revisited the disentanglement of speaker (x-vector), linguistic (BN features) and intonation (pitch) information used in the speaker anonymization scheme described in Section 5.3 so as to analytically bound the speaker information contained in pitch and BN features using DP. Plugging our proposed DP-pitch and BN extractors in a full speaker anonymization pipeline, we showed that we can re-synthesize utterances in a DP fashion. We also empirically showed that our analytical privacy guarantees translate into clear gains in practical privacy guarantees against strong attackers using ASI and ASV approaches, with high intelligibility.

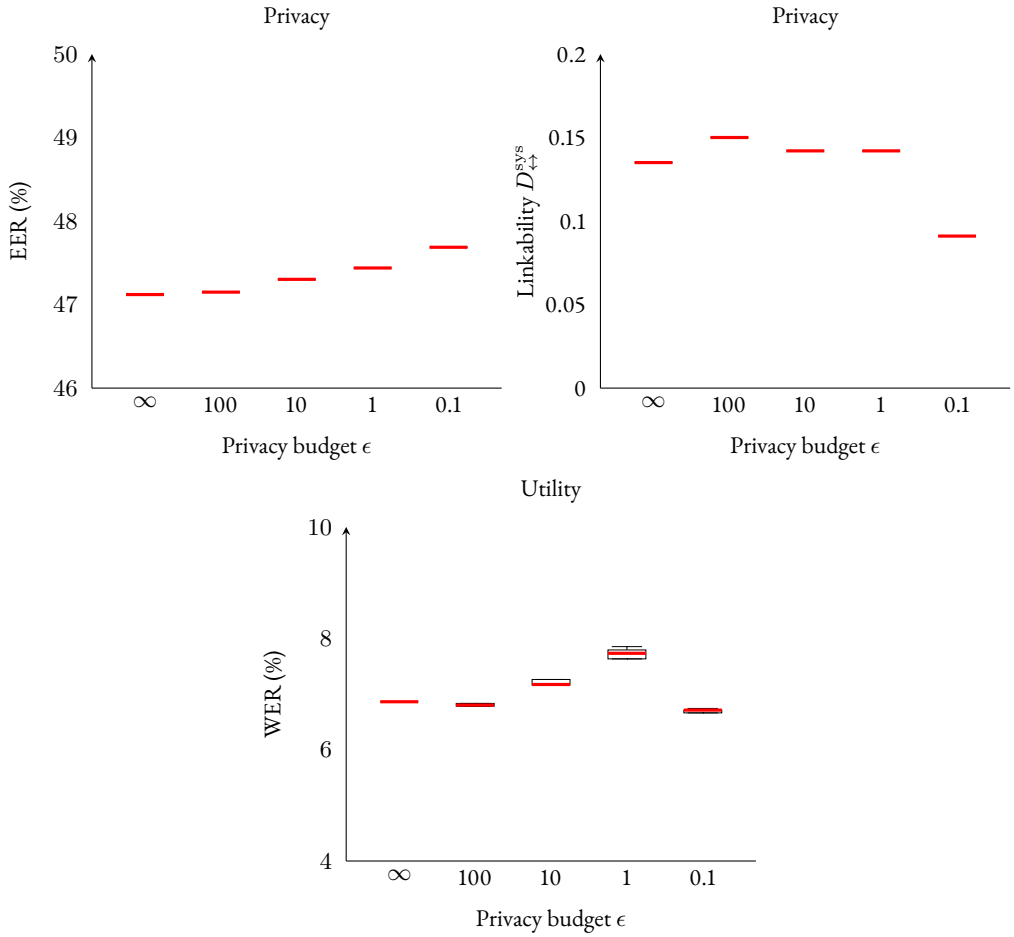


Fig. 6.8 Practical privacy in terms of EER and $D_{\leftrightarrow}^{\text{sys}}$, and utility in terms of WER of `Anon+PC` and `Anon+DP-Pitch` for different ϵ of 100, 10, 1 and 0.1. The ASR_{eval} system for each ϵ is trained on the anonymized utterances synthesized using the BN features of the same ϵ.

Table 6.3 Practical privacy and utility of our `Anon+DP` speech with different analytical privacy budgets against the baseline with pitch conversion `Anon+PC`.

Method	**Privacy**				**Utility**
	Analytical		*Practical*		*Practical*
	BN	Pitch	EER (%)	$D_{\leftrightarrow}^{\text{sys}}$	WER
`Anon+PC`	∞	∞	47.09	0.14	6.8%
`Anon+DP`	100	1.0	48.45	0.11	5.8%
`Anon+DP`	100	0.1	48.99	0.10	5.6%
`Anon+DP`	10	1.0	48.72	0.13	6.5%
`Anon+DP`	10	0.1	48.69	0.13	6.4%
`Anon+DP`	1	1.0	48.72	0.12	7.0%
`Anon+DP`	1	0.1	48.94	0.10	6.7%

Chapter 7

Conclusions and Perspectives

> Arguing that you don't care about the right to privacy because you have nothing to hide is no different than saying you don't care about free speech because you have nothing to say.
>
> *Edward Snowden*

We conclude the thesis by first summarizing the crucial findings and global conclusions, followed by a list of the short-term extensions and the long-term directions that will further strengthen privacy protection and add value to the state-of-the-art anonymization approach. Finally, we present a larger vision for the real-world deployment of the ideas presented in this thesis.

Global summary Recall the central question of this thesis that we asked in Section 1.2:

> *How to effectively remove the biometric identity of the speaker from any speech utterance, while maintaining the usefulness of the signal?*

To genuinely answer this question, in Chapter 3, we first defined a threat model where speakers want to publish their data while protecting their biometric identity, users want to use this data to train machine learning models or conduct research, and attackers want to re-identify the speakers from the published corpus. Within this framework, we proposed a strict evaluation protocol using a continuum of attackers who possess various degrees of knowledge about the anonymization scheme and conducted a comparative analysis of privacy metrics that express different types of vulnerabilities in the anonymized data. A preliminary study with three voice conversion-based anonymization methods revealed that, while the anonymized utterances are somewhat protected against the *Semi-Informed* attackers, the privacy protection provided by established approaches can be completely reversed by the *Informed* attackers. We also found that the global linkability measure can provide complementary information to the EER that is useful for determining the level of protection reliably. At this point, it is important to note that if an attacker fails to reverse the effect of anonymization this does not prove that the anonymization is universally superior. On the contrary, if an attacker succeeds, this proves that the anonymization scheme is suboptimal. In practice, it is infeasible to design all the possible attackers and test our approaches against them. Recall that the previous literature considered only *Ignorant* attackers for evaluating their anonymization schemes, therefore we acknowledge the *Semi-Informed* attacker as a step towards a more realistic adversary and provide empirical results against it. This limitation was eventually lifted by the use of differentially private algorithms in Chapter 6 which

provide a provable analytical lower bound on privacy, thereby corroborating the empirical evidence provided by the attackers.

In Chapter 4, we considered a variant of the above threat model involving the users of digital assistants. We investigated whether there is personally identifiable information in the intermediate representations of an ASR network situated on the manufacturer's cloud which can be used to re-identify the users. We found that there is indeed a significant amount of speaker-related information in such representations, and proposed an adversarial learning-based approach to remove it. Now, the ASR network can be split into a client-side encoder producing private representations, that can then be sent to a server-side decoder for speech-to-text operation. Although such a method would protect the users of digital assistants against re-identification attacks, the approach has certain limitations due to the tight coupling of the encoder and the decoder that are trained together. Moreover, this approach does not allow the publication of anonymized speech samples that have wider usability.

Chapter 5 presented the x-vector based anonymization scheme which was used as the primary baseline for the first VoicePrivacy challenge. It outputs anonymized speech signals which can be published to form an anonymized public corpus. We investigated the design choices for random target selection to increase the resilience of this approach against re-identification attacks and found the best anonymization scheme based on the evaluation against the *Semi-Informed* attackers. We conducted a rigorous evaluation of this scheme in the presence of thousands of speakers and found that, even in the worst case, the anonymization significantly reduces the threat of re-identification as compared to the original speech. We also found that the anonymized corpus is well suited to train a state-of-the-art ASR model when mixed with a small amount of original speech. Although this anonymization scheme gave superior privacy protection than any of the existing approaches, it was based on the assumption that all of the speaker-related information is concentrated only in the x-vector features. We briefly investigated the validity of this assumption and studied the effect of pitch conversion to remove source speaker identity information from the pitch sequence of the utterance. While the Gaussian pitch normalization approach does enhance the privacy of anonymized speech, it occasionally introduces illegal values in the transformed pitch sequence which leads to some loss of utility. Chapter 6 further challenged this "perfect disentanglement" assumption and investigated the pitch as well as the bottleneck features that were used alongside the anonymized x-vector to generate the private speech. We found a significant amount of speaker-related information in pitch as well as BN features, and proposed differential privacy based approach to remove the residual identity markers from the anonymized speech. This approach also allows us to provide analytical bounds and formal privacy guarantees for the proposed anonymization schemes.

Extensions and open problems There is a lot of room for extending the techniques proposed in this thesis. The most immediate extension is to use the adversarial learning based approach proposed in Chapter 4 to remove the speaker-related information from BN features and then plug these new representations either in the "vanilla" x-vector based anonymization scheme (Chapter 5) or the DP version (Chapter 6). We observed that adversarial learning with a single speaker-adversarial branch does not immediately generalize to unseen speakers, hence we plan to use multiple speaker-adversarial branches or to subsume these branches into a single one using Bayesian neural networks [122], such that the speaker-specific attributes could be neutralized from the bottleneck representation. Another useful extension would be to provide a range of user controls that can be used to easily manipulate the anonymization scheme. We attempted to do this by proposing the four design choices in Chapter 5, but it would be interesting to explore more fine-grained attributes, such as the random selection of the target speaker's age group or the target emotion, the ability to change the target speaker over time within each utterance, etc., and verify their potential to anonymize effectively.

The pitch conversion technique we used in this thesis also showed optimistic prospects to remove residual speaker information from the intonation features with some loss of utility as explained before (see Section 5.3.5). A constrained source to target pitch mapping approach is potentially beneficial for enhanced protection as well as the naturalness of the output speech. Additionally, the randomness introduced in the pitch sequence by DP noise addition also needs to be further examined so that the sudden decline in linkability and WER noticed in Figure 6.8 can be explained. This might lead to a superior privacy protection without compromising the utility of anonymized speech.

Another promising extension could be to formulate stronger attack scenarios to validate the resilience of the state-of-the-art anonymization scheme. We emphasize the need for better attacker design as one of our primary proposition for evaluation. Throughout the thesis, we generally consider that the *Semi-Informed* attacker is the most realistic and the strongest adversary to perform re-identification, where the attacker uses the same anonymization scheme as the speaker to anonymize the training set. Briefly, in Chapter 3 (see Table 3.5), we re-examined this consideration and asked the question: how well do the attackers using a given set of anonymization design choices perform on speech samples protected using other design choices? Now we ask the same question in the context of the x-vector based anonymization scheme. Previously, we had observed in Figure 5.10 that the utterance-level assignment (see Section 5.3.2.4) of pseudo-speakers while anonymizing a data set, gives better privacy protection than the speaker-level assignment under the *Lazy-Informed* attack. Due to the constraint mentioned in Section 5.3.4.1, we did not consider the utterance-level assignment for further experimentation. In the course of writing this thesis, we re-visited the utterance-level assignment and trained *Semi-Informed* attackers using this scheme.

We observe that the utterance-level *Semi-Informed* attacker re-identifies speakers in the test set, that is using the same anonymization scheme, with an EER of 13.89%. This is significantly better than the speaker-level *Semi-Informed* attacker which obtains an EER of 41% over the test set anonymized using speaker-level assignment. Furthermore, surprisingly the utterance-level attacker performs similarly over the test set protected using the speaker-level assignment scheme, effectively reducing the best EER from 41% to just 14%, that is a 65% relative loss of privacy protection. We also observed that the EER values for different random seeds in case of speaker-level assignment exhibit higher standard deviation than utterance-level assignment. Nevertheless, this level of protection is substantially better than the previously proposed anonymizations approaches, such as VoiceMask and VTLN-based VC, which provide almost no protection against *Semi-Informed* attacks with an EER of 5–6% (see Table 3.3). Moving forward, this new attack scenario reflects that the privacy protection against a single *Semi-Informed* attacker is not absolute and a full-spectrum analysis is needed to reliably ascertain the strength of anonymization. Consequently, the results obtained in this preliminary experiment reset the baseline for future investigations.

In the long term, it is also desirable to investigate the effect of anonymization over some of the strategic attributes of speech that are of crucial importance to the prospective users of speaker anonymization. As mentioned in Section 1.1, anonymization is essential for anybody using or building a voice interface, but the widespread adoption of this technology requires that the strategic attributes are not distorted. For customer care call-centers it may be the emotional content, for remote health monitoring it may be the pathological conditions, and for educational applications, it may be the disfluencies. Due to such a wide range of speech applications, it is an ongoing effort to investigate the distortion of these attributes and improve the anonymization scheme for seamless usability. Another lesser investigated property of speaker anonymization is its capacity to scale to multiple languages. Throughout the pipeline of the x-vector based anonymization scheme, only the BN features are language-dependent because they are extracted from an ASR network that is trained over English speech. The anonymization itself is evaluated over publicly available English speech corpora, hence it remains to be seen how well it performs over other languages. The long-term goal is to build a multilingual anonymization tool that can be easily scaled to new languages. For that, as a

first step we plan to conduct experiments with BN features extracted from a multilingual ASR network, and possibly also re-train the x-vector extractor over speakers from multiple languages to remove the language bias in the speaker representation.

Beyond research We envisage a larger vision for the research conducted in this thesis and would like to make efforts for the real-world deployment of the speaker anonymization tool such that speech privacy protection becomes accessible to everyone. The first step towards this goal is to prepare a full system of anonymization that can be readily employed by the users. For instance, this thesis focuses only on the removal of the biometric identity of speakers from speech, while several identity markers may also be present in the textual content of the utterance, such as the speaker's name, address, credit card information, etc., which must be redacted out. This is achieved by aligning the textual content with speech using an ASR system followed by an efficient named entity recognizer to identify sensitive patterns. Similar work was done by the partners of the H2020 COMPRISE[1] project which partly funded this thesis. Again, the full system must be highly optimized to provide anonymized utterances in real-time, otherwise the applicability of such a system will be limited to offline operations. Finally, the author of this thesis is dedicated to packaging the software tools developed in this thesis into a deployable form and provide them as a commercial service so that they can be used by society at large and individuals can easily exercise their right to privacy.

[1] https://www.compriseh2020.eu/

References

[1] Abdi, N., Ramokapane, K. M., and Such, J. M. (2019). More than smart speakers: security and privacy perceptions of smart home personal assistants. In *15th Symposium on Usable Privacy and Security*, pages 451–466.

[2] Abowd, J. M. (2018). Protecting the confidentiality of America's statistics: Adopting modern disclosure avoidance methods at the census bureau. *Census Blogs: Research Matters*. `https://www.census.gov/newsroom/blogs/research-matters/2018/08/protecting_the_confi.html`.

[3] Adami, A. G., Mihaescu, R., Reynolds, D. A., and Godfrey, J. J. (2003). Modeling prosodic dynamics for speaker recognition. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages IV–788.

[4] Adams, O., Wiesner, M., Watanabe, S., and Yarowsky, D. (2019). Massively multilingual adversarial speech recognition. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 96–108.

[5] Adi, Y., Zeghidour, N., Collobert, R., Usunier, N., Liptchinsky, V., and Synnaeve, G. (2019). To reverse the gradient or not: An empirical comparison of adversarial and multi-task learning in speech recognition. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3742–3746.

[6] Ahmed, S., Chowdhury, A. R., Fawaz, K., and Ramanathan, P. (2020). Preech: A system for privacy-preserving speech transcription. In *29th USENIX Security Symposium*, pages 2703–2720.

[7] Alexander, A., Botti, F., Dessimoz, D., and Drygajlo, A. (2004). The effect of mismatched recording conditions on human and automatic speaker recognition in forensic applications. *Forensic Science International*, 146:S95–S99.

[8] Algabri, M., Mathkour, H., Bencherif, M. A., Alsulaiman, M., and Mekhtiche, M. A. (2017). Automatic speaker recognition for mobile forensic applications. *Mobile Information Systems*, 2017:1–6.

[9] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (2016). Deep Speech 2 : End-to-end speech recognition in English and Mandarin. In *33rd International Conference on Machine Learning*, pages 173–182.

[10] Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. (2020). Common Voice: A massively-multilingual speech corpus. In *12th Language Resources and Evaluation Conference*, pages 4218–4222.

[11] Arık, S. Ö., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., Sengupta, S., and Shoeybi, M. (2017). Deep Voice: Real-time neural text-to-speech. In *34th International Conference on Machine Learning*, pages 195–204.

[12] Ateniese, G., Mancini, L. V., Spognardi, A., Villani, A., Vitali, D., and Felici, G. (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150.

[13] Ayala-Rivera, V. and Pasquale, L. (2018). The grace period has ended: An approach to operationalize GDPR requirements. In *26th IEEE International Requirements Engineering Conference*, pages 136–146.

[14] Bachan, J., Kuczmarski, T., and Francuzik, P. (2012). Evaluation of synthetic speech using automatic speech recognition. In *XIV International PhD Workshop (OWD 2012). Conference Archives PTETiS*, volume 30, pages 500–505.

[15] Backstrom, L., Dwork, C., and Kleinberg, J. (2007). Wherefore art thou R3579X? Anonymized social networks, hidden patterns, and structural steganography. In *16th International Conference on World Wide Web*, pages 181–190.

[16] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2016). End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4945–4949.

[17] Bahmaninezhad, F., Zhang, C., and Hansen, J. H. (2018). Convolutional neural network based speaker de-identification. In *Odyssey*, pages 255–260.

[18] Ballmer, T. and Brennstuhl, W. (2013). *Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs*, volume 8. Springer Science & Business Media.

[19] Banisar, D. and Davies, S. (1999). Global trends in privacy protection: An international survey of privacy, data protection, and surveillance laws and developments. *The John Marshall Journal of Information Technology & Privacy Law*, 18(1):1.

[20] Battenberg, E., Chen, J., Child, R., Coates, A., Li, Y. G. Y., Liu, H., Satheesh, S., Sriram, A., and Zhu, Z. (2017). Exploring neural transducers for end-to-end speech recognition. In *2017 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 206–213.

[21] Bayardo, R. J. and Agrawal, R. (2005). Data privacy through optimal k-anonymization. In *21st International Conference on Data Engineering*, pages 217–228.

[22] Bayerl, S. P., Brasser, F., Busch, C., Frassetto, T., Jauernig, P., Kolberg, J., Nautsch, A., Riedhammer, K., Sadeghi, A.-R., Schneider, T., Stapf, E., Treiber, A., and Weinert, C. (2019). Privacy-preserving speech processing via STPC and TEEs (poster). In *Privacy Preserving Machine Learning – CCS 2019 Workshop*.

[23] Beenau, B. W., Bonalle, D. S., Fields, S. W., Gray, W. J., Larkin, C., Montgomery, J. L., and Saunders, P. D. (2010). Voiceprint biometrics on a payment device. US Patent 7,814,332.

[24] Beigi, G., Shu, K., Guo, R., Wang, S., and Liu, H. (2019). I am not what I write: Privacy preserving text representation learning. *arXiv preprint arXiv:1907.03189*.

[25] Biemans, M. (1998). The effect of biological gender (sex) and social gender (gender identity) on three pitch measures. *Linguistics in the Netherlands*, 15(1):41–52.

[26] Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., and Reynolds, D. A. (2004). A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004(4):1–22.

[27] Bispham, M. K., Agrafiotis, I., and Goldsmith, M. (2018). A taxonomy of attacks via the speech interface. In *3rd International Conference on CyberTechnologies and CyberSystems*, pages 1–8.

[28] Black, A., Taylor, P., Caley, R., and Clark, R. (1998). The festival speech synthesis system. `http://www.festvox.org/festival/`.

[29] Bohn, D. (2019). Amazon says 100 million Alexa devices have been sold — what's next? `https://www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp`.

[30] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., Van Overveldt, T., Petrou, D., Ramage, D., and Roselander, J. (2019). Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*.

[31] Brasser, F., Frassetto, T., Riedhammer, K., Sadeghi, A.-R., Schneider, T., and Weinert, C. (2018). Voiceguard: Secure and private speech processing. In *Interspeech*, pages 1303–1307.

[32] Brümmer, N. and du Preez, J. A. (2006). Application-independent evaluation of speaker detection. *Computer Speech and Language*, 20(2-3):230–275.

[33] Burkhardt, F. (2005). Emofilt: the simulation of emotional speech by prosody-transformation. In *Interspeech*, pages 509–512.

[34] Campbell, J. P. (1997). Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462.

[35] Carr, P. (2019). *English Phonetics and Phonology: An Introduction*. John Wiley & Sons.

[36] Champion, P., Jouvet, D., and Larcher, A. (2020). Speaker information modification in the VoicePrivacy 2020 toolchain. Technical report. `https://hal.archives-ouvertes.fr/hal-02995855`.

[37] Champion, P., Jouvet, D., and Larcher, A. (2021). A study of F0 modification for x-vector based speech pseudonymization across gender. In *2nd AAAI Workshop on Privacy-Preserving Artificial Intelligence*.

[38] Chatterjee, S. (2019). Is data privacy a fundamental right in india?: An analysis and recommendations from policy and legal perspective. *International Journal of Law and Management*, 61(1):170–190.

[39] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *28th International Conference on Neural Information Processing Systems*, pages 577–585.

[40] Chou, J.-c., Yeh, C.-c., and Lee, H.-y. (2019). One-shot voice conversion by separating speaker and content representations with instance normalization. *arXiv preprint arXiv:1904.05742*.

[41] Chou, J.-c., Yeh, C.-c., Lee, H.-y., and Lee, L.-s. (2018). Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. *arXiv preprint arXiv:1804.02812*.

[42] Chung, H., Iorga, M., Voas, J., and Lee, S. (2017). Alexa, can i trust you? *Computer*, 50(9):100–104.

[43] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *2014 NeurIPS Deep Learning and Representation Learning Workshop*.

[44] Chung, J. S., Nagrani, A., and Zisserman, A. (2018). VoxCeleb2: Deep speaker recognition. In *Interspeech*, pages 1086–1090.

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

[46] Commission Nationale de l'Informatique et des Libertés (2020). "On the record": CNIL publishes a white paper on voice assistants. `https://www.cnil.fr/en/record-cnil-publishes-white-paper-voice-assistants`.

[47] Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., and Wang, T. (2018). Privacy at scale: Local differential privacy in practice. In *2018 International Conference on Management of Data*, pages 1655–1658.

[48] Costan, V. and Devadas, S. (2016). Intel SGX Explained. *International Association for Cryptologic Research — Cryptology ePrint Archive*, 2016(86):1–118.

[49] Coull, S. E., Wright, C. V., Monrose, F., Collins, M. P., and Reiter, M. K. (2007). Playing devil's advocate: Inferring sensitive information from anonymized network traces. In *The Network and Distributed System Security Symposium*, pages 35–47.

[50] Craig, D. W., Pearson, J. V., Szelinger, S., Sekar, A., Redman, M., Corneveaux, J. J., Pawlowski, T. L., Laub, T., Nunn, G., Stephan, D. A., Homer, N., and Huentelman, M. J. (2008). Identification of genetic variants using bar-coded multiplexed sequencing. *Nature Methods*, 5(10):887–893.

[51] Csáji, B. C. (2001). Approximation with artificial neural networks. Master's thesis, Eötvös Loránd University.

[52] Dathathri, R., Saarikivi, O., Chen, H., Laine, K., Lauter, K., Maleki, S., Musuvathi, M., and Mytkowicz, T. (2019). CHET: an optimizing compiler for fully-homomorphic neural-network inferencing. In *40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 142–156.

[53] de Jong, G., McDougall, K., and Nolan, F. (2007). Sound change and speaker identity: an acoustic study. In *Speaker Classification II*, pages 130–141.

[54] Dehak, N., Dumouchel, P., and Kenny, P. (2007). Modeling prosodic features with joint factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2095–2103.

[55] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

[56] Denisov, P., Vu, N. T., and Font, M. F. (2018). Unsupervised domain adaptation by adversarial learning for robust speech recognition. In *13th ITG-Symposium on Speech Communication*, pages 1–5.

[57] Desai, S., Raghavendra, E. V., Yegnanarayana, B., Black, A. W., and Prahallad, K. (2009). Voice conversion using artificial neural networks. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3893–3896.

[58] Deutsch, D., Henthorn, T., and Lapidis, R. (2011). Illusory transformation from speech to song. *The Journal of the Acoustical Society of America*, 129(4):2245–2252.

[59] Di Cerbo, F. and Trabelsi, S. (2018). Towards personal data identification and anonymization using machine learning techniques. In *European Conference on Advances in Databases and Information Systems*, pages 118–126.

[60] Dias, M., Abad, A., and Trancoso, I. (2018). Exploring hashing and cryptonet based approaches for privacy-preserving speech emotion recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2057–2061.

[61] Dibazar, A. A., Narayanan, S., and Berger, T. W. (2002). Feature analysis for automatic detection of pathological speech. In *2nd Joint EMBS-BMES Conference*, volume 1, pages 182–183.

[62] Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2013). Local privacy and statistical minimax rates. In *54th IEEE Annual Symposium on Foundations of Computer Science*, pages 429–438.

[63] Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2018). Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201.

[64] Dudley, H. (1939a). The automatic synthesis of speech. *Proceedings of the National Academy of Sciences of the United States of America*, 25(7):377.

[65] Dudley, H. (1939b). Remaking speech. *The Journal of the Acoustical Society of America*, 11(2):169–177.

[66] Dudley, H. (1940a). The carrier nature of speech. *Bell System Technical Journal*, 19(4):495–515.

[67] Dudley, H. (1940b). The vocoder—electrical re-creation of speech. *Journal of the Society of Motion Picture Engineers*, 34(3):272–278.

[68] Dueck, D. (2009). *Affinity Propagation: Clustering Data by Passing Messages*. PhD thesis, University of Toronto.

[69] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *3rd Theory of Cryptography Conference*, pages 265–284.

[70] Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407.

[71] d'Alessandro, C. (2012). Voice source parameters and prosodic analysis. In *Methods in Empirical Prosody Research*, pages 63–88.

[72] Edu, J. S., Such, J. M., and Suarez-Tangil, G. (2020). Smart home personal assistants: a security and privacy review. *ACM Computing Surveys*, 53(6):1–36.

[73] Eide, E. and Gish, H. (1996). A parametric approach to vocal tract length normalization. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 346–348.

[74] El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587.

[75] Espinoza-Cuadros, F. M., Perero-Codosero, J. M., Antón-Martín, J., and Hernández-Gómez, L. A. (2020). Speaker de-identification system using autoencoders and adversarial training. *arXiv preprint arXiv:2011.04696*.

[76] European Commission (2002). The ePrivacy Directive. `https://edps.europa.eu/data-protection/our-work/subjects/eprivacy-directive_en`.

[77] European Commission (2016). General data protection regulation. `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN`.

[78] European Data Protection Board (2021). Guidelines 02/2021 on virtual voice assistants. `https://edpb.europa.eu/our-work-tools/public-consultations-art-704/2021/guidelines-022021-virtual-voice-assistants_en`.

[79] Fang, F., Wang, X., Yamagishi, J., Echizen, I., Todisco, M., Evans, N., and Bonastre, J.-F. (2019). Speaker anonymization using x-vector and neural waveform models. In *10th ISCA Speech Synthesis Workshop*, pages 155–160.

[80] Fant, G. (1993). Some problems in voice source analysis. *Speech Communication*, 13(1-2):7–22.

[81] Fant, G. (2004). *Speech Acoustics and Phonetics: Selected Writings*, volume 24. Springer Science & Business Media.

[82] Fant, G., Kruckenberg, A., Liljencrants, J., and Båvegård, M. (1994). Voice source parameters in continuous speech, transformation of LF-parameters. In *3rd International Conference on Spoken Language Processing*, pages 1451–1454.

[83] Farrus, M., Wagner, M., Anguita, J., and Hernando, J. (2008). How vulnerable are prosodic features to professional imitators? In *Odyssey*, pages 1–6.

[84] Fernandes, E., Jung, J., and Prakash, A. (2016). Security analysis of emerging smart home applications. In *2016 IEEE Symposium on Security and Privacy*, pages 636–654.

[85] Feutry, C., Piantanida, P., Bengio, Y., and Duhamel, P. (2018). Learning anonymized representations with adversarial neural networks. *arXiv preprint arXiv:1802.09386*.

[86] Feyisetan, O., Balle, B., Drake, T., and Diethe, T. (2020). Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *13th International Conference on Web Search and Data Mining*, pages 178–186.

[87] Fienberg, S. E., Rinaldo, A., and Yang, X. (2010). Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *International Conference on Privacy in Statistical Databases*, pages 187–199.

[88] Fontan, L., Ferrané, I., Farinas, J., Pinquier, J., Tardieu, J., Magnen, C., Gaillard, P., Aumont, X., and Füllgrabe, C. (2017). Automatic speech recognition predicts speech intelligibility and comprehension for listeners with simulated age-related hearing loss. *Journal of Speech, Language, and Hearing Research*, 60(9):2394–2405.

[89] Forney, G. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

[90] Fu, S.-W., Li, P.-C., Lai, Y.-H., Yang, C.-C., Hsieh, L.-C., and Tsao, Y. (2016). Joint dictionary learning-based non-negative matrix factorization for voice conversion to improve speech intelligibility after oral surgery. *IEEE Transactions on Biomedical Engineering*, 64(11):2584–2594.

[91] Fung, B. C., Wang, K., Chen, R., and Yu, P. S. (2010). Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 42(4):1–53.

[92] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

[93] Garson, J. (2010). Connectionism. *Stanford Encyclopedia of Philosophy*. `https://stanford.library.usyd.edu.au/archives/sum2010/entries/connectionism/`.

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

[95] Ghahremani, P., Manohar, V., Hadian, H., Povey, D., and Khudanpur, S. (2017). Investigation of transfer learning for ASR using LF-MMI trained neural networks. In *2017 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 279–286.

[96] Glackin, C., Chollet, G., Dugan, N., Cannings, N., Wall, J., Tahir, S., Ray, I. G., and Rajarajan, M. (2017). Privacy preserving encrypted phonetic search of speech data. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6414–6418.

[97] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *13th International Conference on Artificial Intelligence and Statistics*, pages 249–256.

[98] Gomez-Barrero, M., Galbally, J., Rathgeb, C., and Busch, C. (2017). General framework to evaluate unlinkability in biometric template protection systems. *IEEE Transactions on Information Forensics and Security*, 13(6):1406–1420.

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

[100] Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772.

[101] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

[102] Gu, Y., Li, X., Chen, S., Zhang, J., and Marsic, I. (2017). Speech intention classification with multimodal deep learning. In *Canadian Conference on Artificial Intelligence*, pages 260–271.

[103] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. In *Interspeech*, pages 5036–5040.

[104] Gupta, O. and Raskar, R. (2018). Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8.

[105] Gupta, P., Prajapati, G. P., Singh, S., Kamble, M. R., and Patil, H. A. (2020). Design of voice privacy system using linear prediction. In *2020 APSIPA Annual Summit and Conference*, pages 543–549.

[106] Gussenhoven, C. (2004). *The Phonology of Tone and Intonation*. Cambridge University Press.

[107] Haderlein, T., Moers, C., Möbius, B., Rosanowski, F., and Nöth, E. (2011). Intelligibility rating with automatic speech recognition, prosodic, and cepstral evaluation. In *International Conference on Text, Speech and Dialogue*, pages 195–202.

[108] Hadian, H., Sameti, H., Povey, D., and Khudanpur, S. (2018). End-to-end speech recognition using lattice-free MMI. In *Interspeech*, pages 12–16.

[109] Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., and Wu, Y. (2020a). ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context. In *Interspeech*, pages 3610–3614.

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

[111] Hannun, A. Y., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

[112] Hardcastle, W. J., Laver, J., and Gibbon, F. E. (2012). *The Handbook of Phonetic Sciences*. John Wiley & Sons.

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

[114] Hedelin, P. (1981). A tone oriented voice excited vocoder. In *1981 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 205–208.

[115] Hellbernd, N. and Sammler, D. (2016). Prosody conveys speaker's intentions: Acoustic cues for speech act perception. *Journal of Memory and Language*, 88:70–86.

[116] Hermansky, H. (2011). Speech recognition from spectral dynamics. *Sadhana*, 36(5):729–744.

[117] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[118] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735 – 1780.

[119] Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2016). Voice conversion from non-parallel corpora using variational auto-encoder. In *2016 APSIPA Annual Summit and Conference*, pages 1–6.

[120] Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2017a). Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*.

[121] Hsu, W.-N., Zhang, Y., and Glass, J. (2017b). Unsupervised learning of disentangled and interpretable representations from sequential data. In *31st International Conference on Neural Information Processing Systems*, pages 1876–1887.

[122] Hu, S., Lam, M. W., Xie, X., Liu, S., Yu, J., Wu, X., Liu, X., and Meng, H. (2019a). Bayesian and Gaussian process neural networks for large vocabulary continuous speech recognition. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6555–6559.

[123] Hu, S., Xie, X., Liu, S., Lam, M. W., Yu, J., Wu, X., Liu, X., and Meng, H. (2019b). LF-MMI training of Bayesian and Gaussian process time delay neural networks for speech recognition. In *Interspeech*, pages 2793–2797.

[124] Huang, W.-C., Hayashi, T., Watanabe, S., and Toda, T. (2020a). The sequence-to-sequence baseline for the Voice Conversion Challenge 2020: Cascading ASR and TTS. *arXiv preprint arXiv:2010.02434*.

[125] Huang, W.-C., Hayashi, T., Wu, Y.-C., Kameoka, H., and Toda, T. (2021). Pretraining techniques for sequence-to-sequence voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:745–755.

[126] Huang, W.-C., Luo, H., Hwang, H.-T., Lo, C.-C., Peng, Y.-H., Tsao, Y., and Wang, H.-M. (2020b). Unsupervised representation disentanglement using cross domain features and adversarial learning in variational autoencoder based voice conversion. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4):468–479.

[127] Huang, X., Baker, J., and Reddy, R. (2014). A historical perspective of speech recognition. *Communications of the ACM*, 57(1):94–103.

[128] Huang, Y., Obada-Obieh, B., and Beznosov, K. (2020c). Amazon vs. my brother: How users of shared smart speakers perceive and cope with privacy risks. In *2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

[129] Ioffe, S. (2006). Probabilistic linear discriminant analysis. In *9th European Conference on Computer Vision*, pages 531–542.

[130] ISO/IEC (2011a). 24745:2011 Information Technology—Security techniques—Biometric Information Protection. https://www.iso.org/standard/52946.html.

[131] ISO/IEC (2011b). 29100:2011 Information Technology—Security techniques—Privacy framework. https://www.iso.org/standard/45123.html.

[132] ISO/IEC (2018). 30136:2018 Information Technology—Performance Testing of Biometric Protection Schemes. https://www.iso.org/standard/53256.html.

[133] ISO/IEC (2021). 19795-1:2021 Information Technology — Biometric performance testing and reporting — Part 1: Principles and framework. https://www.iso.org/standard/73515.html.

[134] ITU-T (1994). Recommendation P.85 (06/94): Telephone transmission quality subjective opinion tests. A method for subjective performance assessment of the quality of speech voice output devices. https://www.itu.int/rec/T-REC-P.85-199406-I/en.

[135] Jackson, C. and Orebaugh, A. (2018). A study of security and privacy issues associated with the Amazon Echo. *International Journal of Internet of Things and Cyber-Assurance*, 1(1):91–100.

[136] Jagielski, M., Ullman, J., and Oprea, A. (2020). Auditing differentially private machine learning: How private is private SGD? In *Advances in Neural Information Processing Systems*, pages 22205–22216.

[137] Jain, A., Hong, L., and Pankanti, S. (2000). Biometric identification. *Communications of the ACM*, 43(2):90–98.

[138] Jayaraman, B. and Evans, D. (2019). Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium*, pages 1895–1912.

[139] Jin, Q., Toth, A. R., Schultz, T., and Black, A. W. (2009). Speaker de-identification via voice transformation. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 529–533.

[140] Juang, B. H. and Chen, T. (1998). The past, present, and future of speech processing. *IEEE Signal Processing Magazine*, 15(3):24–48.

[141] Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Pearson Education.

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

[143] Juvela, L., Bollepalli, B., Yamagishi, J., and Alku, P. (2019). GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram. *arXiv preprint arXiv:1904.03976*.

[144] Kameoka, H., Kaneko, T., Tanaka, K., and Hojo, N. (2018). StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks. In *2018 IEEE Spoken Language Technology Workshop*, pages 266–273.

[145] Kaneko, T. and Kameoka, H. (2018). CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks. In *26th European Signal Processing Conference*, pages 2100–2104.

[146] Kanervisto, A., Vestman, V., Sahidullah, M., Hautamäki, V., and Kinnunen, T. (2017). Effects of gender information in text-independent and text-dependent speaker verification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5360–5364.

[147] Karmakar, P., Teng, S. W., and Lu, G. (2021). Thank you for attention: A survey on attention-based artificial neural networks for automatic speech recognition. *arXiv preprint arXiv:2102.07259*.

[148] Kasi, K. (2002). Yet another algorithm for pitch tracking (yaapt). Master's thesis, Old Dominion University.

[149] Kawahara, H. (2006). STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds. *Acoustical Science and Technology*, 27(6):349–353.

[150] Kenny, P. (2010). Bayesian speaker verification with heavy-tailed priors. In *Odyssey*, pages 14–21.

[151] Kepuska, V. and Bohouta, G. (2018). Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In *8th IEEE Annual Computing and Communication Workshop and Conference*, pages 99–103.

[152] Kim, C. and Stern, R. M. (2008). Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In *Interspeech*, pages 2598–2601.

[153] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization (poster). In *International Conference on Learning Representations*.

[154] Kinnunen, T. and Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40.

[155] Kiparsky, P. (2003). The phonological basis of sound change. *The Handbook of Historical Linguistics*, pages 311–342.

[156] Kniesza, V. (1988). Sound substitution, sound change, spelling in French loanwords in Middle English. In *Luick Revisited: Papers Read at the Luick-Symposium at Schloss Liechtenstein*, pages 205–220.

[157] Ko, T., Peddinti, V., Povey, D., Seltzer, M. L., and Khudanpur, S. (2017). A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5220–5224.

[158] Kopparapu, S. K. and Laxminarayana, M. (2010). Choice of Mel filter bank in computing MFCC of a resampled speech. In *10th International Conference on Information Science, Signal Processing and their Applications*, pages 121–124.

[159] Kottasová, I. (2018). These companies are getting killed by GDPR. *CNN Business*. `https://money.cnn.com/2018/05/11/technology/gdpr-tech-companies-losers/index.html`.

[160] Kotti, M. and Kotropoulos, C. (2008). Gender classification in two emotional speech databases. In *19th International Conference on Pattern Recognition*, pages 1–4.

[161] Kratzenstein, C. G. (1782). Sur la naissance de la formation des voyelles. *Journal de Physique*, 21:358–380.

[162] Kwon, O.-W., Chan, K., Hao, J., and Lee, T.-W. (2003). Emotion recognition by speech signals. In *EuroSpeech*, pages 125–128.

[163] Labov, W. (1963). The social motivation of a sound change. *Word*, 19(3):273–309.

[164] Ladefoged, P. (1996). *Elements of Acoustic Phonetics*. University of Chicago Press.

[165] Latif, S., Khalifa, S., Rana, R., and Jurdak, R. (2020). Federated learning for speech emotion recognition applications. In *19th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 341–342.

[166] Lau, J., Zimmerman, B., and Schaub, F. (2018). Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–31.

[167] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[168] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[169] Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. (2019). Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy*, pages 656–672.

[170] Lee, H., Kim, S., Kim, J. W., and Chung, Y. D. (2017). Utility-preserving anonymization for health data publishing. *BMC Medical Informatics and Decision Making*, 17(1):1–12.

[171] Lei, X., Tu, G.-H., Liu, A. X., Li, C.-Y., and Xie, T. (2017). The insecurity of home digital voice assistants — Amazon Alexa as a case study. *arXiv preprint arXiv:1712.03327*.

[172] Leong, R. (2018). Analyzing the privacy attack landscape for Amazon Alexa devices. Technical report, Imperial College London.

[173] Leroy, D., Coucke, A., Lavril, T., Gisselbrecht, T., and Dureau, J. (2019). Federated learning for keyword spotting. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6341–6345.

[174] Levin, A. and Nicholson, M. J. (2005). Privacy law in the United States, the EU and Canada: The allure of the middle ground. *University of Ottawa Law & Technology Journal*, 2:357.

[175] Li, H., Tu, M., Huang, J., Narayanan, S., and Georgiou, P. (2020). Speaker-invariant affective representation learning via adversarial training. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7144–7148.

[176] Li, N., Li, T., and Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115.

[177] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., and He, B. (2019). A survey on federated learning systems: vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693*.

[178] Liao, C.-F., Tsao, Y., Lee, H.-Y., and Wang, H.-M. (2018). Noise adaptive speech enhancement using domain adversarial training. *arXiv preprint arXiv:1807.07501*.

[179] Lieberman, P., Laitman, J. T., Reidenberg, J. S., and Gannon, P. J. (1992). The anatomy, physiology, acoustics and perception of speech: essential elements in analysis of the evolution of human speech. *Journal of Human Evolution*, 23(6):447–467.

[180] Lin, J.-L. and Wei, M.-C. (2008). An efficient clustering method for k-anonymization. In *2008 International Workshop on Privacy and Anonymity in Information Society*, pages 46–50.

[181] Lippmann, R. P. (1989). Review of neural networks for speech recognition. *Neural Computation*, 1(1):1–38.

[182] Liu, K., Zhang, J., and Yan, Y. (2007). High quality voice conversion through phoneme-based linear mapping functions with STRAIGHT for Mandarin. In *4th International Conference on Fuzzy Systems and Knowledge Discovery*, volume 4, pages 410–414.

[183] López, G., Quesada, L., and Guerrero, L. A. (2017). Alexa vs. Siri vs. Cortana vs. Google Assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics*, pages 241–250.

[184] Lorenzo-Trueba, J., Fang, F., Wang, X., Echizen, I., Yamagishi, J., and Kinnunen, T. (2018). Can we steal your vocal identity from the internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data. In *Odyssey*, pages 240–247.

[185] Luger, E. and Sellen, A. (2016). "Like having a really bad PA" The gulf between user expectation and experience of conversational agents. In *2016 CHI Conference on Human Factors in Computing Systems*, pages 5286–5297.

[186] Lukács, A. (2016). What is privacy? The history and definition of privacy. In *Tavaszi Szél Tanulmánykötet I*, pages 256–265. http://publicatio.bibl.u-szeged.hu/10794/7/3188699.pdf.

[187] Luong, H.-T. and Yamagishi, J. (2019). Bootstrapping non-parallel voice conversion from speaker-adaptive text-to-speech. In *2019 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 200–207.

[188] Lyu, L., He, X., and Li, Y. (2020). Differentially private representation for NLP: Formal guarantee and an empirical study on privacy and fairness. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2355–2365.

[189] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1):3–54.

[190] Magariños, C., Lopez-Otero, P., Docio-Fernandez, L., Rodriguez-Banga, E., Erro, D., and Garcia-Mateo, C. (2017). Reversible speaker de-identification using pre-trained transformation functions. *Computer Speech and Language*, 46:36–52.

[191] Malkin, N., Deatrick, J., Tong, A., Wijesekera, P., Egelman, S., and Wagner, D. (2019). Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019(4):250–271.

[192] Manohar, V. (2019). *Semi-Supervised Training for Automatic Speech Recognition*. PhD thesis, Johns Hopkins University.

[193] Maouche, M., Srivastava, B. M. L., Vauquier, N., Bellet, A., Tommasi, M., and Vincent, E. (2020). A comparative study of speech anonymization metrics. In *Interspeech*, pages 1708–1712.

[194] Mary, L. and Yegnanarayana, B. (2008). Extraction and representation of prosodic features for language and speaker recognition. *Speech Communication*, 50(10):782–796.

[195] Matrouf, D., Bonastre, J.-F., and Fredouille, C. (2006). Effect of speech transformation on impostor acceptance. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, pages 933–936.

[196] Mawalim, C. O., Galajit, K., Karnjana, J., and Unoki, M. (2020). X-vector singular value modification and statistical-based decomposition with ensemble regression modeling for speaker anonymization system. In *Interspeech*, pages 1703–1707.

[197] McAdams, S. E. (1984). *Spectral Fusion, Spectral Parsing and the Formation of Auditory Images*. PhD thesis, Stanford University.

[198] McAulay, R. and Quatieri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754.

[199] Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., and Bengio, Y. (2016). SampleRNN: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*.

[200] Meng, Z., Li, J., Chen, Z., Zhao, Y., Mazalov, V., Gong, Y., and Juang, B.-H. (2018a). Speaker-invariant training via adversarial learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5969–5973.

[201] Meng, Z., Li, J., and Gong, Y. (2018b). Adversarial feature-mapping for speech enhancement. *arXiv preprint arXiv:1809.02251*.

[202] Mercuri, R. T. and Neumann, P. G. (2003). Security by obscurity. *Communications of the ACM*, 46(11):160.

[203] Miao, Y., Gowayyed, M., and Metze, F. (2015). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 167–174.

[204] Morales, N., Gu, L., and Gao, Y. (2007). Adding noise to improve noise robustness in speech recognition. In *Interspeech*, pages 930–933.

[205] Morise, M. (2015). CheapTrick, a spectral envelope estimator for high-quality speech synthesis. *Speech Communication*, 67:1–7.

[206] Morise, M., Yokomori, F., and Ozawa, K. (2016). WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, 99(7):1877–1884.

[207] Muntés-Mulero, V. and Nin, J. (2009). Privacy and anonymization for very large datasets. In *18th ACM Conference on Information and Knowledge Management*, pages 2117–2118.

[208] Nagrani, A., Chung, J. S., and Zisserman, A. (2017). VoxCeleb: A large-scale speaker identification dataset. In *Interspeech*, pages 2616–2620.

[209] Nakashika, T., Takiguchi, T., and Ariki, Y. (2014). Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):580–587.

[210] Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy*, pages 111–125.

[211] Nasr, M., Shokri, R., and Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy*, pages 739–753.

[212] Nasr, M., Song, S., Thakurta, A., Papernot, N., and Carlini, N. (2021). Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy*, pages 866–882.

[213] Nautsch, A., Jasserand, C., Kindt, E., Todisco, M., Trancoso, I., and Evans, N. (2019a). The GDPR & speech data: Reflections of legal and technology communities, first steps towards a common understanding. *arXiv preprint arXiv:1907.03458*.

[214] Nautsch, A., Jiménez, A., Treiber, A., Kolberg, J., Jasserand, C., Kindt, E., Delgado, H., Todisco, M., Hmani, M. A., Mtibaa, A., Abdelraheem, M. A., Abad, A., Teixeira, F., Matrouf, D., Gomez-Barrero, M., Petrovska-Delacrétaz, D., Chollet, G., Evans, N., Schneider, T., Bonastre, J.-F., Raj, B., Trancoso, I., and Busch, C. (2019b). Preserving privacy in speaker and speech characterisation. *Computer Speech & Language*, 58:441–480.

[215] Neekhara, P., Hussain, S., Dubnov, S., Koushanfar, F., and McAuley, J. (2021). Expressive neural voice cloning. *arXiv preprint arXiv:2102.00151*.

[216] Ning, Y., He, S., Wu, Z., Xing, C., and Zhang, L.-J. (2019). A review of deep learning based speech synthesis. *Applied Sciences*, 9(19):4050.

[217] Oppenheim, A. V., Buck, J. R., and Schafer, R. W. (2001). *Discrete-time Signal Processing*. Prentice Hall.

[218] Orlandi, C., Piva, A., and Barni, M. (2007). Oblivious neural network computing via homomorphic encryption. *EURASIP Journal on Information Security*, 2007:1–11.

[219] Paliwal, K. and Wójcicki, K. (2008). Effect of analysis window duration on speech intelligibility. *IEEE Signal Processing Letters*, 15:785–788.

[220] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5206–5210.

[221] Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. (2021). Tempered sigmoid activations for deep learning with differential privacy. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 9312–9321.

[222] Pathak, M. A. (2012). *Privacy-Preserving Machine Learning for Speech Processing*. PhD thesis, Carnegie Mellon University.

[223] Patino, J., Tomashenko, N., Todisco, M., Nautsch, A., and Evans, N. (2020). Speaker anonymisation using the McAdams coefficient. *arXiv preprint arXiv:2011.01130*.

[224] Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Interspeech*, pages 3214–3218.

[225] Perry, T. L., Ohde, R. N., and Ashmead, D. H. (2001). The acoustic bases for gender identification from children's voices. *The Journal of the Acoustical Society of America*, 109(6):2988–2998.

[226] Peskin, B., Navratil, J., Abramson, J., Jones, D., Klusacek, D., Reynolds, D. A., and Xiang, B. (2003). Using prosodic and conversational features for high-performance speaker recognition: report from JHU WS'02. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 792–795.

[227] Phong, L. T., Aono, Y., Hayashi, T., Wang, L., and Moriai, S. (2018). Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345.

[228] Pobar, M. and Ipšić, I. (2014). Online speaker de-identification using voice transformation. In *37th International Convention on Information and Communication Technology, Electronics and Microelectronics*, pages 1264–1267.

[229] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747.

[230] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*.

[231] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for asr based on lattice-free MMI. In *Interspeech*, pages 2751–2755.

[232] PrivacyGuard (2019). Smart speaker technology: Privacy risks and solutions. `https://blog.privacyguard.com/post/smart-speaker-technology-privacy-risks-and-solutions`.

[233] Qian, J., Du, H., Hou, J., Chen, L., Jung, T., and Li, X.-Y. (2018). Hidebehind: Enjoy voice input with voiceprint unclonability and anonymity. In *16th ACM Conference on Embedded Networked Sensor Systems*, pages 82–94.

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

[235] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[236] Raj, D., Snyder, D., Povey, D., and Khudanpur, S. (2019). Probing the information encoded in x-vectors. In *2019 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 726–733.

[237] Ramos, D. (2018). Voice assistants: How artificial intelligence assistants are changing our lives every day. `https://www.smartsheet.com/voice-assistants-artificial-intelligence`.

[238] Ramteke, P. B., Dixit, A. A., Supanekar, S., Dharwadkar, N. V., and Koolagudi, S. G. (2018). Gender identification from children's speech. In *2018 International Conference on Contemporary Computing*, pages 1–6.

[239] Rane, S. and Boufounos, P. T. (2013). Privacy-preserving nearest neighbor methods: Comparing signals without revealing them. *IEEE Signal Processing Magazine*, 30(2):18–28.

[240] Rasmussen, D. J. (2013). Voice print identification for identifying speakers. US Patent 8,606,579.

[241] Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17(1-2):91–108.

[242] Rohdin, J., Biswas, S., and Shinoda, K. (2014). Constrained discriminative PLDA training for speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1670–1674.

[243] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386.

[244] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

[245] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[246] Ruggiero, G., Zovato, E., Di Caro, L., and Pollet, V. (2021). Voice cloning: a multi-speaker text-to-speech synthesis approach based on transfer learning. *arXiv preprint arXiv:2102.05630*.

[247] Rumelhart, D. E. (1989). *Parallel Distributed Processing*. MIT Press.

[248] Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jégou, H. (2019). White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567.

[249] Salmun, I., Opher, I., and Lapidot, I. (2016). On the use of PLDA i-vector scoring for clustering short segments. In *Odyssey*, pages 407–414.

[250] Samarati, P. and Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International.

[251] Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 55–59.

[252] Schuller, B. and Batliner, A. (2013). *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. John Wiley & Sons.

[253] Schuller, B., Steidl, S., Batliner, A., Nöth, E., Vinciarelli, A., Burkhardt, F., van Son, R., Weninger, F., Eyben, F., Bocklet, T., Mohammadi, G., and Weiss, B. (2015). A survey on perceived speaker traits: Personality, likability, pathology, and the first challenge. *Computer Speech & Language*, 29(1):100–131.

[254] Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., and Kim, S. (2013). The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In *Interspeech*, pages 148–152.

[255] Sekiyama, K. (1997). Cultural and linguistic factors in audiovisual speech processing: The McGurk effect in Chinese subjects. *Perception & Psychophysics*, 59(1):73–80.

[256] Serdyuk, D., Audhkhasi, K., Brakel, P., Ramabhadran, B., Thomas, S., and Bengio, Y. (2016). Invariant representations for noisy speech recognition. *arXiv preprint arXiv:1612.01928*.

[257] Shamsabadi, A. S., Teixeira, F. S., Abad, A., Raj, B., Cavallaro, A., and Trancoso, I. (2021). FoolHD: Fooling speaker identification by highly imperceptible adversarial disturbances. In *46th IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6159–6163, Toronto, Canada.

[258] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., Saurous, R. A., Agiomvrgiannakis, Y., and Wu, Y. (2018). Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4779–4783.

[259] Shi, J., Amith, J. D., Castillo García, R., Guadalupe Sierra, E., Duh, K., and Watanabe, S. (2021). Leveraging end-to-end ASR for endangered language documentation: An empirical study on yolóxochitl Mixtec. In *16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1134–1145.

[260] Shinohara, Y. (2016). Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Interspeech*, pages 2369–2372.

[261] Sholokhov, A., Kinnunen, T., Vestman, V., and Lee, K. A. (2020a). Extrapolating false alarm rates in automatic speaker verification. In *Interspeech*, pages 4218–4222.

[262] Sholokhov, A., Kinnunen, T., Vestman, V., and Lee, K. A. (2020b). Voice biometrics security: Extrapolating false alarm rate via hierarchical Bayesian modeling of speaker verification scores. *Computer Speech & Language*, 60:1–19.

[263] Shon, S., Dehak, N., Reynolds, D., and Glass, J. (2019). MCE 2018: The 1st multi-target speaker detection and identification challenge evaluation. In *Interspeech*, pages 356–360.

[264] Signol, F., Barras, C., and Lienard, J.-S. (2008). Evaluation of the Pitch Estimation Algorithms in the monopitch and multipitch cases. *The Journal of the Acoustical Society of America*, 123(5):3077–3081.

[265] Sisman, B., Yamagishi, J., King, S., and Li, H. (2021). An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:132–157.

[266] Smaragdis, P. and Shashanka, M. (2007). A framework for secure speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1404–1413.

[267] Smith, C. L. (2000). *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press.

[268] Snyder, D., Chen, G., and Povey, D. (2015). MUSAN: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484v1*.

[269] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5329–5333.

[270] Spanias, A. S. (1994). Speech coding: A tutorial review. *Proceedings of the IEEE*, 82(10):1541–1582.

[271] Sriram, A., Jun, H., Gaur, Y., and Satheesh, S. (2018). Robust speech recognition using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5639–5643.

[272] Srivastava, B. M. L., Bellet, A., Tommasi, M., and Vincent, E. (2019). Privacy-preserving adversarial representation learning in ASR: Reality or Illusion? In *Interspeech*, pages 3700–3704.

[273] Srivastava, B. M. L., Vauquier, N., Sahidullah, M., Bellet, A., Tommasi, M., and Vincent, E. (2020). Evaluating voice conversion-based privacy protection against informed attackers. In *2020 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2802–2806.

[274] Stevens, K. N. (2000). *Acoustic Phonetics*. MIT press.

[275] Stolcke, A., Shriberg, E., Bates, R., Coccaro, N., Jurafsky, D., Martin, R., Meteer, M., Ries, K., Taylor, P., and Van Ess-Dykema, C. (1998). Dialog act modeling for conversational speech. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105.

[276] Stylianou, Y. (2009). Voice transformation: a survey. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3585–3588.

[277] Sun, L., Li, K., Wang, H., Kang, S., and Meng, H. (2016). Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *2016 IEEE International Conference on Multimedia and Expo*, pages 1–6.

[278] Sun, S., Yeh, C.-F., Hwang, M.-Y., Ostendorf, M., and Xie, L. (2018). Domain adversarial training for accented speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4854–4858.

[279] Sundermann, D. and Ney, H. (2003). VTLN-based voice conversion. In *3rd IEEE International Symposium on Signal Processing and Information Technology*, pages 556–559.

[280] Sweeney, L. (2000). Simple demographics often identify people uniquely. *Health (San Francisco)*, 671(2000):1–34.

[281] Synnaeve, G., Xu, Q., Kahn, J., Likhomanenko, T., Grave, E., Pratap, V., Sriram, A., Liptchinsky, V., and Collobert, R. (2019). End-to-end ASR: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*.

[282] Székely, É., Henter, G. E., Beskow, J., and Gustafson, J. (2019). Spontaneous conversational speech synthesis from found data. In *Interspeech*, pages 4435–4439.

[283] Talkin, D. (1995). A Robust Algorithm for Pitch Tracking (RAPT). In *Speech Coding and Synthesis*, volume 495, pages 495–518.

[284] Tan, H. H. and Lim, K. H. (2019). Vanishing gradient mitigation with deep learning neural network optimization. In *7th International Conference on Smart Computing & Communications*, pages 1–4.

[285] Tan, Z.-H. and Dehak, N. (2020). rVAD: An unsupervised segment-based robust voice activity detection method. *Computer speech & language*, 59:1–21.

[286] Tanaka, K., Kameoka, H., Kaneko, T., and Hojo, N. (2019). AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6805–6809.

[287] Taylor, J. and Richmond, K. (2020). Enhancing sequence-to-sequence text-to-speech with morphology. In *Interspeech*, pages 1738–1742.

[288] Teixeira, F., Abad, A., and Trancoso, I. (2019). Privacy-preserving paralinguistic tasks. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6575–6579.

[289] Thiruvaran, T., Ambikairajah, E., and Epps, J. (2008). FM features for automatic forensic speaker recognition. In *Interspeech*, pages 1497–1500.

[290] Tian, X., Chng, E. S., and Li, H. (2019). A speaker-dependent WaveNet for voice conversion with non-parallel data. In *Interspeech*, pages 201–205.

[291] Tian, X., Wang, J., Xu, H., Chng, E. S., and Li, H. (2018). Average modeling approach to voice conversion with non-parallel data. In *Odyssey*, pages 227–232.

[292] Toda, T., Black, A. W., and Tokuda, K. (2007). Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235.

[293] Tomashenko, N., Srivastava, B. M. L., Wang, X., Vincent, E., Nautsch, A., Yamagishi, J., Evans, N., Bonastre, J.-F., Noé, P.-G., Todisco, M., and Patino, J. (2020a). The VoicePrivacy 2020 Challenge evaluation plan. `https://www.voiceprivacychallenge.org/docs/VoicePrivacy_2020_Eval_Plan_v1_3.pdf`.

[294] Tomashenko, N., Srivastava, B. M. L., Wang, X., Vincent, E., Nautsch, A., Yamagishi, J., Evans, N., Patino, J., Bonastre, J.-F., Noé, P.-G., and Todisco, M. (2020b). Introducing the VoicePrivacy initiative. In *Interspeech*, pages 1693–1697.

[295] Tomashenko, N., Wang, X., Vincent, E., Patino, J., Srivastava, B. M. L., Noé, P.-G., Nautsch, A., Evans, N., Yamagishi, J., O'Brien, B., Chanclu, A., Bonastre, J.-F., Todisco, M., and Maouche, M. (2021). The VoicePrivacy 2020 Challenge: Results and findings. *arXiv preprint arXiv:2109.00648*.

[296] Torra, V. and Navarro-Arribas, G. (2016). Big data privacy and anonymization. In *IFIP International Summer School on Privacy and Identity Management*, pages 15–26.

[297] Tripathi, A., Mohan, A., Anand, S., and Singh, M. (2018). Adversarial learning of raw speech features for domain invariant speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5959–5963.

[298] Tsuchiya, T., Tawara, N., Ogawa, T., and Kobayashi, T. (2018). Speaker invariant feature extraction for zero-resource languages with adversarial learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2381–2385.

[299] Tu, M., Tang, Y., Huang, J., He, X., and Zhou, B. (2019). Towards adversarial learning of speaker-invariant representation for speech emotion recognition. *arXiv preprint arXiv:1903.09606*.

[300] Turner, H., Lovisotto, G., and Martinovic, I. (2020). Speaker anonymization with distribution-preserving x-vector generation for the VoicePrivacy Challenge 2020. *arXiv preprint arXiv:2010.13457*.

[301] Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Conference on Computer Vision and Pattern Recognition*, pages 6924–6932.

[302] Umapathy, K. and Krishnan, S. (2005). Feature analysis of pathological speech signals using local discriminant bases technique. *Medical and Biological Engineering and Computing*, 43(4):457–464.

[303] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

[304] van Leeuwen, D. A. and Brümmer, N. (2007). An introduction to application-independent evaluation of speaker recognition systems. In *Speaker Classification I: Fundamentals, Features, and Methods*, pages 330–353.

[305] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

[306] Ververidis, D. and Kotropoulos, C. (2004). Automatic speech classification to five emotional states based on gender information. In *12th European Signal Processing Conference*, pages 341–344.

[307] Vestman, V., Kinnunen, T., Hautamäki, R. G., and Sahidullah, M. (2020). Voice mimicry attacks assisted by automatic speaker verification. *Computer Speech & Language*, 59:36–54.

[308] Villani, C. (2009). *Optimal Transport: Old and New*. Springer.

[309] Vinciarelli, A., Pantic, M., and Bourlard, H. (2009). Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759.

[310] von Kempelen, W. (1791). *Le Mécanisme de la Parole, Suivi de la Description d'Une Machine Parlante*. B. Bauer.

[311] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

[312] Wang, X., Takaki, S., and Yamagishi, J. (2019a). Neural source-filter-based waveform model for statistical parametric speech synthesis. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5916–5920.

[313] Wang, X., Takaki, S., and Yamagishi, J. (2019b). Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415.

[314] Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017). Tacotron: A fully end-to-end text-to-speech synthesis model. *arXiv preprint arXiv:1703.10135*, 164.

[315] Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., and Saurous, R. A. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189.

[316] Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.

[317] Warren, S. D. and Brandeis, L. D. (1890). Right to privacy. *Harvard Law Review*, 4:193–220.

[318] Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., and Ochiai, T. (2018). ESPnet: End-to-end speech processing toolkit. In *Interspeech*, pages 2207–2211.

[319] Watanabe, S., Hori, T., Kim, S., Hershey, J. R., and Hayashi, T. (2017). Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

[320] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

[321] Wheatstone, C. (2011). *The Scientific Papers of Sir Charles Wheatstone*. Cambridge University Press.

[322] Wiesner, M., Renduchintala, A., Watanabe, S., Liu, C., Dehak, N., and Khudanpur, S. (2018). Pretraining by backtranslation for end-to-end ASR in low-resource settings. *arXiv preprint arXiv:1812.03919*.

[323] won Park, S., young Kim, D., and chul Joe, M. (2020). Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data. In *Interspeech*, pages 4696–4700.

[324] Wu, Y.-C., Hwang, H.-T., Hsu, C.-C., Tsao, Y., and Wang, H.-M. (2016). Locally linear embedding for exemplar-based spectral conversion. In *Interspeech*, pages 1652–1656.

[325] Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., and Li, H. (2015). Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130–153.

[326] Wu, Z., Virtanen, T., Chng, E. S., and Li, H. (2014). Exemplar-based sparse representation with residual compensation for voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1506–1521.

[327] Yi, J., Tao, J., Wen, Z., and Bai, Y. (2018). Language-adversarial transfer learning for low-resource speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(3):621–630.

[328] Yin, Y., Huang, B., Wu, Y., and Soleymani, M. (2020). Speaker-invariant adversarial domain adaptation for emotion recognition. In *2020 International Conference on Multimodal Interaction*, pages 481–490.

[329] Yoo, I.-C., Lee, K., Leem, S., Oh, H., Ko, B., and Yook, D. (2020). Speaker anonymization for personal information protection using voice conversion techniques. *IEEE Access*, 8:198637–198645.

[330] Yu, D. and Deng, L. (2016). *Automatic Speech Recognition.* Springer.

[331] Yu, D. and Seltzer, M. (2011). Improved bottleneck features using pretrained deep neural networks. In *Interspeech*, pages 237–240.

[332] Zahorian, S. A. and Hu, H. (2008). A spectral/temporal method for robust fundamental frequency tracking. *The Journal of the Acoustical Society of America*, 123(6):4559–4571.

[333] Zeghidour, N., Xu, Q., Liptchinsky, V., Usunier, N., Synnaeve, G., and Collobert, R. (2018). Fully convolutional speech recognition. *arXiv preprint arXiv:1812.06864*.

[334] Zeiler, M., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., and Hinton, G. (2013). On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521.

[335] Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R. J., Jia, Y., Chen, Z., and Wu, Y. (2019). LibriTTS: A corpus derived from LibriSpeech for text-to-speech. pages 1526–1530.

[336] Zeng, Y.-M., Wu, Z.-Y., Falk, T., and Chan, W.-Y. (2006). Robust GMM based gender classification using pitch and RASTA-PLP parameters of speech. In *International Conference on Machine Learning and Cybernetics*, pages 3376–3379.

[337] Zhang, J.-X., Ling, Z.-H., Jiang, Y., Liu, L.-J., Liang, C., and Dai, L.-R. (2019a). Improving sequence-to-sequence voice conversion by adding text-supervision. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6785–6789.

[338] Zhang, M., Sisman, B., Zhao, L., and Li, H. (2020). DeepConversion: Voice conversion with limited parallel training data. *Speech Communication*, 122:31–43.

[339] Zhang, M., Wang, X., Fang, F., Li, H., and Yamagishi, J. (2019b). Joint training framework for text-to-speech and voice conversion using multi-source Tacotron and WaveNet. *arXiv preprint arXiv:1903.12389*.

[340] Zhang, M., Zhou, Y., Zhao, L., and Li, H. (2021). Transfer learning from speech synthesis to voice conversion with non-parallel training data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1290–1302.

[341] Zhang, S.-X., Gong, Y., and Yu, D. (2019c). Encrypted speech recognition using deep polynomial networks. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5691–5695.

[342] Zhang, Y., Chan, W., and Jaitly, N. (2017). Very deep convolutional networks for end-to-end speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4845–4849.

[343] Zhao, L., Mammadov, M., and Yearwood, J. (2010). From convex to nonconvex: a loss function analysis for binary classification. In *2010 IEEE International Conference on Data Mining Workshops*, pages 1281–1288.

[344] Zhou, Y., Tian, X., Xu, H., Das, R. K., and Li, H. (2019). Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6790–6794.

Appendix A

Supplementary Results for Large-scale Speaker Study

A.1 Gender identification in Mozilla Common Voice

A large number of speakers in the Common Voice data set did not specify their gender, which is crucial for conducting trials pertaining to the experiments described in Section 5.4. Before training a gender identification method over the training set composed of speakers who specified their gender, we visualized the x-vector space of the training set as presented in Fig. A.1. We notice that there is a significant overlap between the male and female clusters which is uncommon in x-vector space as observed in other data sets. After manually listening to some of the outlier audio samples we discovered that this overlap is due to a large amount of *children speakers* and to *gender mislabeling*. Previous studies [225, 238] have shown that identifying gender in the presence of children's voices is especially challenging. We also observe that the

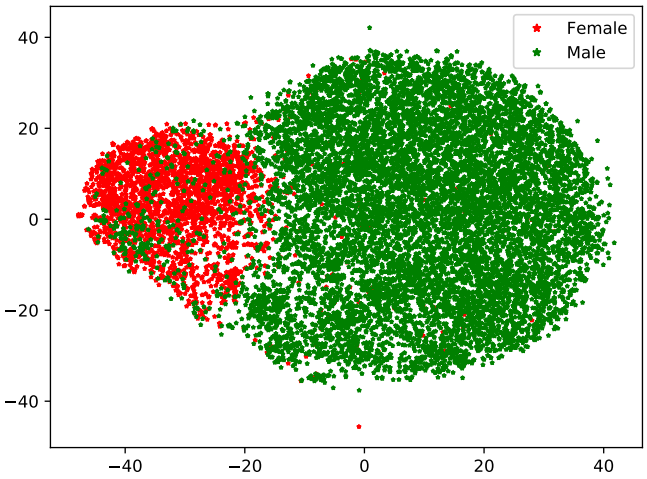
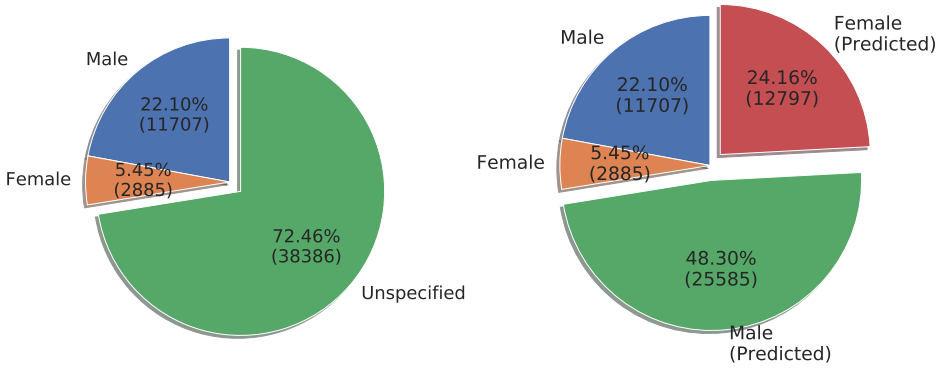


Fig. A.1 t-SNE representation of speaker x-vectors in the Common Voice data set.

labelled part of the data set consists of a large number of male speakers (22.1%) and only a small amount of female speakers (5.45%). This imbalance would result in huge bias against female speakers. For an unbiased gender identification, we employ the technique suggested by Kanervisto et al. [146], where the x-vectors are first projected into 1-D space using linear discriminant analysis (LDA) and then clustered using Gaussian

mixture models (GMM). We obtain an F-1 score of 92% for female speakers and 91% for male speakers in the test set. The speaker gender distribution before and after gender identification is presented in Fig. A.2.



(a) Before prediction (b) After prediction

Fig. A.2 Speaker gender distribution observed in the Common Voice data set. The exact number of speakers for each gender are indicated in parentheses. Four speakers are discarded due to lack of data.

We notice that out of the 72.46% speakers with unspecified gender, 48.3% of the speakers are predicted to be male and only 24.16% as female. Moreover, many children speakers are classified as female hence the number of female speakers is further reduced. Since there is a large number of confirmed male speakers in the overall data set, we choose to conduct the large-scale speaker study in Section 5.4 only with the male speakers.

A.2 Worst-case analysis: extra results

In this appendix, we present some extra results for the worst-case analysis of the anonymization scheme performed in Section 5.4.5. First, we analyze the normalized rank of the overall worst-performing utterance in the CV-trial set (see Table 5.10) as shown in Figure A.3. As expected, the normalized rank for the worst-performing utterance is always close to zero, i.e., the speakers of this utterance can be re-identified with close to 100% accuracy in baseline and *Semi-Informed* case. There is a slight improvement in protection in *Ignorant* and *Lazy-Informed* case, but it not significant with respect to privacy.

Next, we look at the worst-performing speaker who exhibits the lowest rank among the 20 trial speakers considered for re-identification in the CV-trial set. The results are shown in Figure A.4. Again, without anonymization the worst-performing speaker can be re-identified with 100% accuracy, but after anonymization in the *Semi-Informed* case, which is the best case for the attacker, the normalized rank improves significantly (≈ 0.1) as compared to the worst-performing utterance (≈ 0). Given the high standard deviation observed in the figure, we can infer that different utterances of this speaker behave quite differently, and several utterances of this speaker are so well protected as if the attacker is operating in *Lazy-Informed* setting. Further investigation is needed to identify the properties of such utterances that make them resilient to *Semi-Informed* attack in the worst case.

Finally, we analyze the single worst-performing utterance of each speaker in the CV-trial set. The results are shown in Figure A.5 and there are two major observations. First, the overall performance in *Semi-Informed* case (> 0.1) is better than the worst-performing speaker and of course, better than the overall worst-performing utterance, so a majority of utterances are quite well protected. The standard deviation is

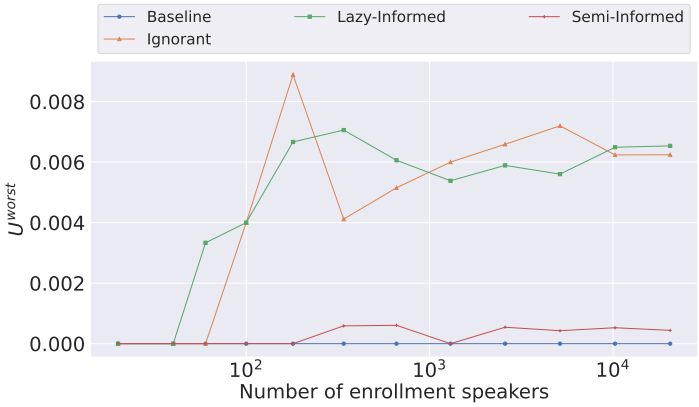


Fig. A.3 Normalized rank for the worst-performing utterance as a function of the enrollment speaker population.

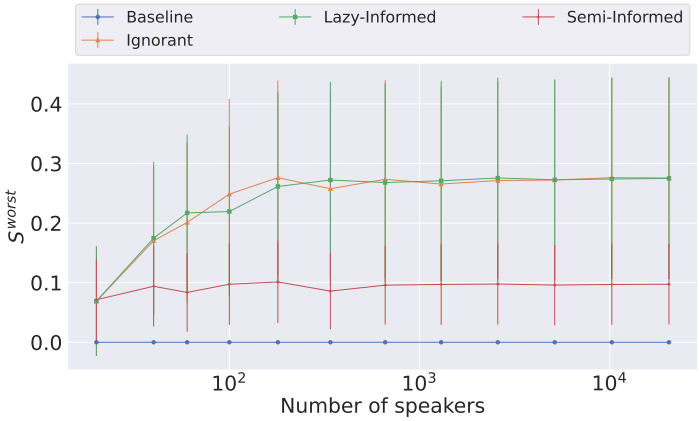


Fig. A.4 Normalized rank of all the utterances from the worst-performing speaker as a function of the enrollment speaker population. Whiskers indicate the standard deviation of the normalized rank for the utterance of the worst-performing speaker.

again quite high, where some utterances are as worse as the baseline without anonymization, while some are so well protected as if the attacker is operating in the *Ignorant* setting. Hence, anonymization works extremely well for some speakers even in the worst-case.

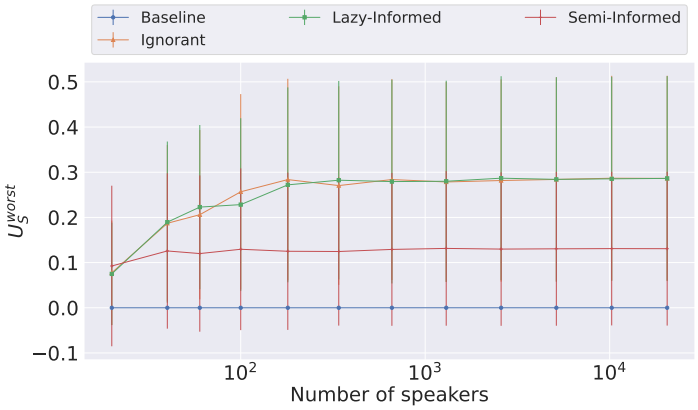


Fig. A.5 Normalized rank of the worst-performing utterance from each speaker in the trial set as a function of the enrollment speaker population. Whiskers indicate the standard deviation of the normalized rank for the worst-performing utterance of each speaker in the trial data set.

Appendix B

Differentially Private ASR Acoustic Modeling

B.1 Gradients for the noise layer $\mathcal{N}$

As shown in Figure B.1, the noise layer $\mathcal{N}$ is composed of an ℓ_1-normalization step followed by a noise addition step, and finally another ℓ_1-normalization step. Let $\mathbf{b} = [b_1, \ldots, b_M]$ be the input and $\bar{\mathbf{b}} = \frac{\mathbf{b}}{\|\mathbf{b}\|_1} = [\bar{b}_1, \ldots, \bar{b}_M]$ the output of the first ℓ_1-normalization step, where $\|\mathbf{b}\|_1 = \sum_m |b_m|$.

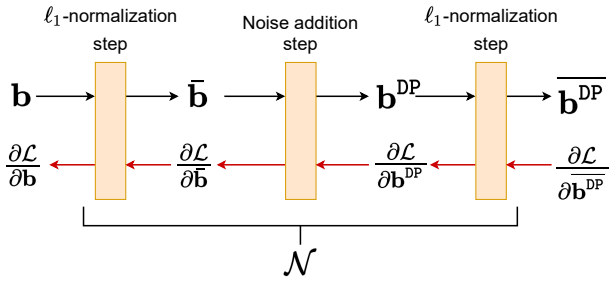


Fig. B.1 Input, output and gradients that pass through the noise layer $\mathcal{N}$.

The gradient of the first ℓ_1-normalization step is computed as follows. The ℓ_1-normalization operation can be written as:

$$\bar{b}_m = \frac{b_m}{\sum_i |b_i|}. \tag{B.1}$$

The individual components of the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{b}}$ are written as:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b_i} &= \sum_m \frac{\partial \mathcal{L}}{\partial \bar{b}_m} \frac{\partial \bar{b}_m}{\partial b_i} \\ &= \frac{\partial \mathcal{L}}{\partial \bar{b}_i} \frac{\partial \bar{b}_i}{\partial b_i} + \sum_{m \neq i} \frac{\partial \mathcal{L}}{\partial \bar{b}_m} \frac{\partial \bar{b}_m}{\partial b_i}. \end{aligned} \tag{B.2}$$

Now, we solve Equation (B.2) independently for $m = i$ and $m \neq i$.

For $m = i$:

$$\begin{aligned}
\frac{\partial \bar{b}_i}{\partial b_i} &= \frac{\partial}{\partial b_i}\left(\frac{b_i}{|b_i| + \sum_{i' \neq i} |b_{i'}|}\right) \\
&= \frac{\sum_{i' \neq i} |b_{i'}|}{(|b_i| + \sum_{i' \neq i} |b_{i'}|)^2} \qquad \text{since} \quad \frac{\partial}{\partial x}\left(\frac{x}{|x| + c}\right) = \frac{c}{(|x| + c)^2} \\
&= \frac{\|\mathbf{b}\|_1 - |b_i|}{\|\mathbf{b}\|_1^2} \\
&= \frac{1 - \frac{|b_i|}{\|\mathbf{b}\|_1}}{\|\mathbf{b}\|_1} \\
&= \frac{1 - |\bar{b}_i|}{\|\mathbf{b}\|_1}.
\end{aligned} \tag{B.3}$$

For $m \neq i$:

$$\begin{aligned}
\frac{\partial \bar{b}_m}{\partial b_i} &= \frac{\partial}{\partial b_i}\left(\frac{b_m}{|b_i| + \sum_{i' \neq i} |b_{i'}|}\right) \\
&= -\frac{b_m \times \operatorname{sign}(b_i)}{(|b_i| + \sum_{i' \neq i} |b_{i'}|)^2} \qquad \text{since} \quad \frac{\partial}{\partial x}\left(\frac{a}{|x| + c}\right) = \frac{a \times \operatorname{sign}(x)}{(|x| + c)^2} \\
&= -\frac{b_m \times \operatorname{sign}(b_i)}{\|\mathbf{b}\|_1^2} \\
&= -\frac{\frac{b_m}{\|\mathbf{b}\|_1} \times \operatorname{sign}(b_i)}{\|\mathbf{b}\|_1} \\
&= -\frac{\bar{b}_m \times \operatorname{sign}(b_i)}{\|\mathbf{b}\|_1}.
\end{aligned} \tag{B.4}$$

Equations (B.3) and (B.4) are combined to get the gradient

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \mathbf{J} \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{b}}}, \tag{B.5}$$

where each element of the Jacobian matrix $\mathbf{J}$ is given by

$$J_{im} = \begin{cases} \frac{1 - |\bar{b}_i|}{\|\mathbf{b}\|_1} & \text{if } i = m, \\ -\frac{\bar{b}_m \times \operatorname{sign}(b_i)}{\|\mathbf{b}\|_1} & \text{otherwise.} \end{cases} \tag{B.6}$$

For implementation purposes we define a new variable κ_{im} as

$$\kappa_{im} = \bar{b}_m \times \operatorname{sign}(b_i). \tag{B.7}$$

Now we want to re-write $\mathbf{J}$ in terms of κ_{im}, so we first derive J_{ii}. From Equation (B.6) we have

$$\begin{aligned} J_{ii} &= \frac{1 - |\bar{b}_i|}{\|\mathbf{b}\|_1} \\ &= \frac{1}{\|\mathbf{b}\|_1}(1 - \bar{b}_i \times \text{sign}(\bar{b}_i)) \\ &= \frac{1}{\|\mathbf{b}\|_1}(1 - \bar{b}_i \times \text{sign}(b_i)) \qquad \text{since} \quad \text{sign}(\bar{b}_i) = \text{sign}(b_i) \\ &= \frac{1 - \kappa_{ii}}{\|\mathbf{b}\|_1} \qquad \text{using Eq. (B.7).} \end{aligned} \tag{B.8}$$

Hence, Equation (B.6) can now be re-written as

$$J_{im} = \begin{cases} \frac{1-\kappa_{im}}{\|\mathbf{b}\|_1} & \text{if } i = m, \\ -\frac{\kappa_{im}}{\|\mathbf{b}\|_1} & \text{otherwise.} \end{cases} \tag{B.9}$$

The two cases can be combined in matrix form as

$$\mathbf{J} = \frac{1}{\|\mathbf{b}\|_1}(\mathbf{I} - \mathbf{K}), \tag{B.10}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{K} = \bar{\mathbf{b}} \otimes \text{sign}(\mathbf{b})$, where $\text{sign}(\mathbf{b}) = [\text{sign}(b_1), \ldots, \text{sign}(b_M)]$ and $\otimes$ represents the outer product of two vectors.

The noise addition step does not affect the gradient:

$$\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{b}}} = \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{\text{DP}}}. \tag{B.11}$$

Indeed, the noise is additive and independent of $\bar{\mathbf{b}}$. Similarly, for the last ℓ_1-normalization step, we have $\overline{\mathbf{b}^{\text{DP}}} = \frac{\mathbf{b}^{\text{DP}}}{\|\mathbf{b}^{\text{DP}}\|_1}$, and the gradients backpropagated from this layer can be derived similar to Equation (B.5), i.e.,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{\text{DP}}} = \mathbf{J}^{\text{DP}} \frac{\partial \mathcal{L}}{\partial \overline{\mathbf{b}^{\text{DP}}}}. \tag{B.12}$$

Now, $\mathbf{J}^{\text{DP}}$ is computed similar to Equation (B.10):

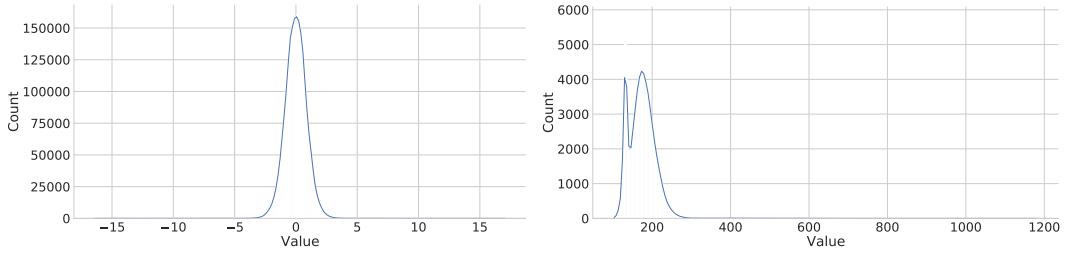
$$\mathbf{J}^{\text{DP}} = \frac{1}{\|\mathbf{b}^{\text{DP}}\|_1}(\mathbf{I} - \mathbf{K}^{\text{DP}}), \tag{B.13}$$

where $\mathbf{K}^{\text{DP}} = \overline{\mathbf{b}^{\text{DP}}} \otimes \text{sign}(\mathbf{b}^{\text{DP}})$.

B.2 Effect of noise layers on ASR bottleneck features

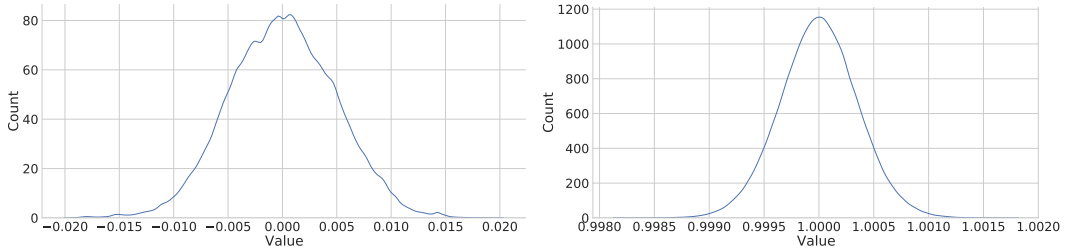
Chapter 6 investigates the presence of speaker-related information in the pitch and BN features extracted from a speech signal. A technique inspired from differential privacy was formulated to remove the residual information from the pitch and the BN features. In this section, we specifically focus on the BN features, where a fixed amount of Laplace noise, depending on the chosen ϵ, is added to these features to make them ϵ-DP. Before adding the noise, the sensitivity of the BN features is fixed by their ℓ_1-normalization as explained

in Section B.1. In this section, we plot the distribution of individual BN features of the test enrollment data (see Table 5.3), and the distribution of the resulting ℓ_1-norm.



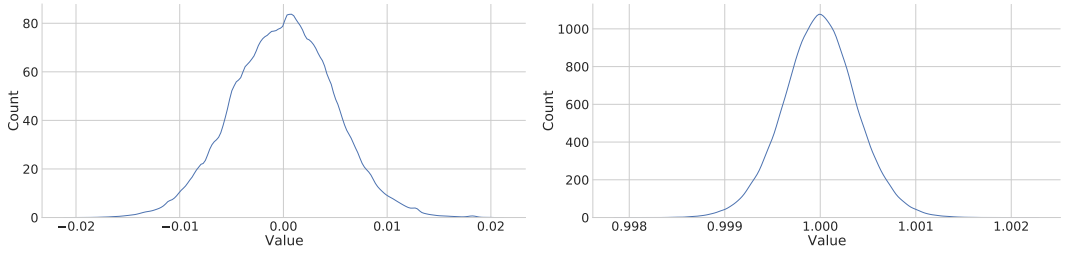
(a) Entries of $\mathbf{b}$ (b) ℓ_1-norm of $\mathbf{b}$

Fig. B.2 Distribution of component values and ℓ_1-norm of the original BN features.



(a) Entries of $\overline{\mathbf{b}^{\text{DP}}}$ (b) ℓ_1-norm of $\overline{\mathbf{b}^{\text{DP}}}$

Fig. B.3 Distribution of the BN features and the resulting ℓ_1-norm at the output of the noise layer, where $\epsilon = 1$.

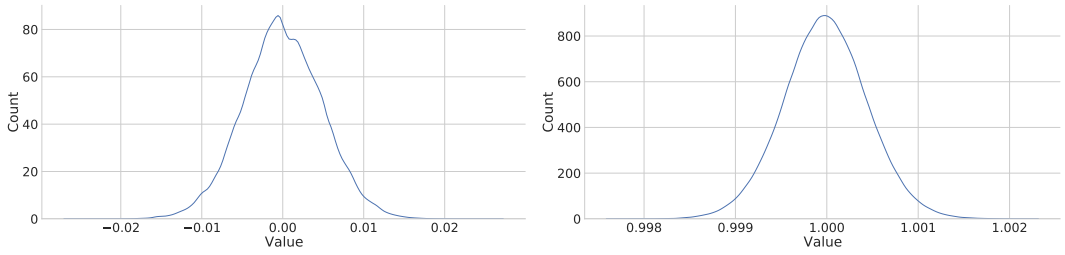


(a) Entries of $\overline{\mathbf{b}^{\text{DP}}}$ (b) ℓ_1-norm of $\overline{\mathbf{b}^{\text{DP}}}$

Fig. B.4 Distribution of the BN features and the resulting ℓ_1-norm at the output of the noise layer, where $\epsilon = 10$.

Figure B.2 shows that the value of each original BN feature is bounded within $[-5, 5]$. The resulting ℓ_1-norm follows a bi-modal[1] distribution, with peaks around 100 and 200, and is unbounded. After the two normalization steps as shown in Figures B.3, B.4 and B.5, the range of feature values becomes much smaller, i.e., $[-0.02, 0.02]$, and the resulting ℓ_1-norm is normally distributed around 1.0 with a very small variance. While this situation is appropriate for noise addition as the sensitivity of the normalized features

[1] exhibiting two peak values



(a) Entries of $\overline{\mathbf{b}^{\text{DP}}}$ (b) ℓ_1-norm of $\overline{\mathbf{b}^{\text{DP}}}$

Fig. B.5 Distribution of the BN features and the resulting ℓ_1-norm at the output of the noise layer, where $\epsilon = 100$.

is bounded, it slightly reduces the expressivity of the ASR network parameters leading to a small loss of generalization and utility, also observed by the increase in WER.

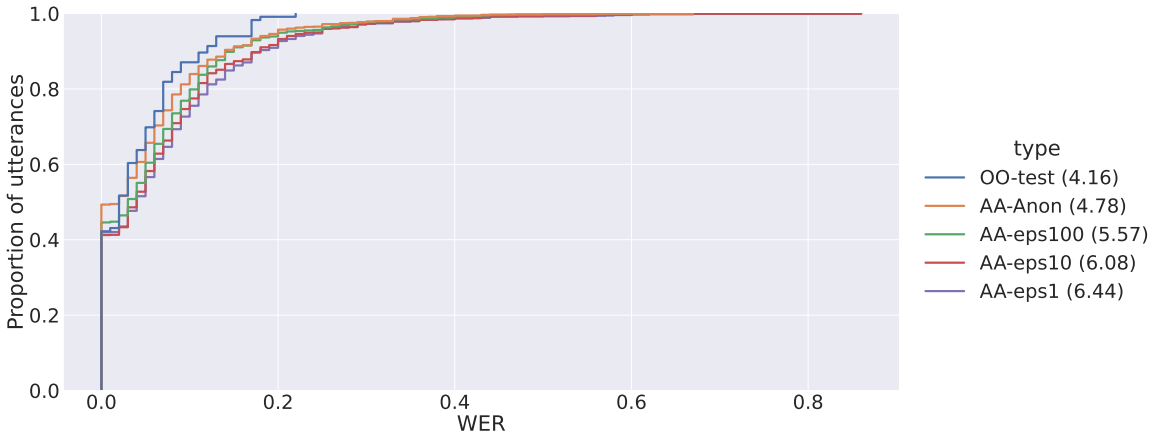


Fig. B.6 Empirical cumulative distribution functions of the WER on x-axis and the corresponding proportion of utterances on y-axis. The blue curve shows the baseline ASR performance. The numbers in brackets in the legend indicate the overall WER (%) over the test data set.

Figure B.6 shows the WER distribution over all the trial utterances in the test data set (Ref. Table 5.3) for speech utterances generated by different BN features. The blue curve (`OO-test`) shows the baseline performance over original speech which is indeed the best performing in terms of utility. The curve shows that a majority of utterances exhibit very low values of WER, and the highest proportion among all the other systems to have zero WER. The next best system, i.e., the orange curve (`AA-Anon`) shows the WER over the utterances anonymized using the original BN features and the anonymization schemes proposed in Section 5.3.[2] The selected design choices are: {*PLDA* distance, *dense* proximity, *random* gender-selection, and *speaker-level* assignment}. The prefix "`AA`" indicates that the ASR model was trained and tested over anonymized utterances which were anonymized using the same anonymization scheme. This, in practice, is the best case decoding scenario for improving the WER over anonymized data.

The green (`AA-eps100`), red (`AA-eps10`) and purple (`AA-eps1`) curves indicate the utility performance of the speech generated using the noise layer $\mathcal{N}$ just after the original BN features. Although, the performance degrades gradually as the value of ϵ decreases, the WER is not unreasonably high even for $\epsilon = 1$. It can be

[2] This scheme is also mentioned in Table 6.2

observed that a relatively higher number of utterances exhibit high WER after anonymization, which reaches 80% for some utterances. Further investigation is needed to identify the cause of such selective increase in the WER.