Design Patterns for Detecting and Mitigating Bias in Edge AI

Hutiri, Wiebke

DOI
10.4233/uuid:c74cc90a-e55d-489c-b3bb-4c8c4a6dd7e6

Publication date
2023

Document Version
Final published version

Citation (APA)
Hutiri, W. (2023). *Design Patterns for Detecting and Mitigating Bias in Edge AI*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:c74cc90a-e55d-489c-b3bb-4c8c4a6dd7e6

Wiebke Hutiri

Design Patterns for Detecting and Mitigating Bias in Edge AI

Design Patterns for Detecting and Mitigating Bias in Edge AI

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Wednesday 1 of November 2023 at 10:00 o'clock

by

Wiebke HUTIRI

Master of Science in Computer Science, University of Cape Town, South Africa
born in Pretoria, South Africa

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. M.F.W.H.A. Janssen	Delft University of Technology, promotor
Dr. Y. Ding	Delft University of Technology, copromotor

Independent members:

Prof. dr. M.E. Warnier	Delft University of Technology
Prof. dr. ir. N. Meratnia	Eindhoven University of Technology
Prof. dr. C.M. Jonker	Delft University of Technology
Prof. dr. J.A. Crowcroft	University of Cambridge, United Kingdom
Prof. Dr. C. Becker	University of Stuttgart, Germany

Keywords: Edge AI, Edge Intelligence, Trustworthy AI, Responsible AI Design, Bias, Fairness, Design Patterns, Speech Technology, Speaker Verification, Keyword Spotting

Printed by: Gildeprint

Cover by: Wiebke Hutiri

ISBN 978-94-6419-932-1

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

In the architectural pattern language there is, at root, behind the whole thing, a constant preoccupation with the question, "Under what circumstances is the environment good"? In architecture that means something. It means something important and vital that goes, ultimately, to the nature of human life...

I understand that the software patterns, insofar as they refer to objects and programs and so on, can make a program better. That isn't the same thing, because in that sentence "better" could mean merely technically efficient, not actually "good." So, I have no idea whether the search for something that helps human life is a formal part of what you are searching for. Or are you primarily searching for—what should I call it—good technical performance? This seems to me a very, very vital issue.

Christopher Alexander,
excerpt from the Keynote Speech to the 1996 OOPSLA Convention

... For those things we call artifacts are not apart from nature. They have no dispensation to ignore or violate natural law. At the same time they are adapted to human goals and purposes. They are what they are in order to satisfy our desire to fly or to eat well. As our aims change, so too do our artifacts and vice versa.

Herbert Simon,
excerpt from The Sciences of the Artificial, p. 3

Contents

Summary

Motivation

From smart phones to speakers and watches, Edge AI is deployed on billions of devices to process large volumes of personal data efficiently, privately and in real-time. With a simple set of keywords, such as "call for help", an Edge AI system can activate important services, like emergency response, with a convenient voice command. While Edge AI applications are promising, many recent incidents of bias in AI systems caution that Edge AI too may systematically discriminate against groups of people based on their gender, race, age, accent, nationality and other personal attributes. More so, as the physical restrictions of Edge AI, together with the complexity of its heterogeneous and decentralised operating environment, pose trade-offs when deploying AI to the edge. This thesis is motivated by the societal demand for trustworthy AI, by the propensity of AI systems to be biased, and consequently by the need to detect and mitigate bias in diverse Edge AI applications. To address this need, generalisable approaches are required to support bias detection and mitigation during Edge AI development.

Objective

This thesis aims to develop design patterns for detecting and mitigating bias in the development of Edge AI systems. The design patterns present a generalisable approach for capturing established practices for bias detection and mitigation in machine learning (ML) so that this knowledge is readily accessible to researchers and practitioners that develop Edge AI, but who have limited experience with detecting and mitigating bias. Moreover, the patterns should offer a tool for researchers and practitioners to share new knowledge and insights about bias in Edge AI between application domains.

Outline and Results

Chapter 1 motivates the need to detect and mitigate bias in Edge AI enabled smart systems. The chapter states the scientific gaps that this research aims to address, and outlines the research goals and questions. Furthermore, the design science research approach and thesis outline are presented.

Chapter 2 introduces technical background knowledge that contextualises this thesis in Edge AI, a computing paradigm for processing personal and sensitive sensor data by extending ML to the periphery of the Internet of Things through edge computing. On-device ML, a category of Edge AI, and the application domain of voice-activated Edge AI are described in further detail.

Chapter 3 moves beyond the technical foundations of Edge AI to related work on trustworthiness, fairness and bias in ML. The chapter highlights that bias detection and mitigation in Edge AI development have been understudied in the literature. It further reviews current approaches to detecting and mitigating bias in ML, and raises implementation and knowledge transfer challenges that limit the adoption of these approaches in Edge AI. The chapter concludes with a review of design patterns as an intervention to overcoming these challenges.

Chapter 4 finds design patterns to detect and mitigate bias in ML from established knowledge and practices. The chapter proposes a conceptual model for guiding pattern discovery, identifies 106 pattern instances in software tools, consolidates them into 23 patterns and organises the patterns in a catalogue. Finally, the chapter formulates two recipes that use a collection of patterns to guide bias measurement and benchmark dataset curation processes.

Chapter 5 validates the utility of the proposed patterns and recipes in a ML use case that investigates speaker verification systems. The chapter validates the utility of the *Benchmark Dataset* and *Bias Measurement* recipes for detecting bias in pretrained models and existing evaluation benchmarks. It then identifies pitfalls in bias evaluations that can be mitigated by using patterns and recipes. The chapter introduces two new patterns that were discovered in the use case.

Chapter 6 extends and adapts the patterns to an Edge AI setting. The chapter is based on the first scientific study of bias in on-device ML, and uses the patterns to investigate the impact of model training and optimisation design choices in a keyword-spotting (KWS) use case. The chapter captures the insights from the bias evaluation as new patterns, thus adapting the pattern catalogue to bias detection and mitigation in on-device ML.

Chapter 7 presents a concluding overview of the research. The chapter discusses the findings, highlights the research contributions, reflects on the limitations and proposes recommendations for future work.

Chapter 8 reflects on the relevance of this thesis beyond its immediate contribution, given the current moment in time. The chapter calls for a design perspective on AI risks and harms, positions the need for increased design and systems thinking in research on trustworthy AI, and finally calls for research on trustworthy AI that extends the system boundaries of AI beyond the technical to the practical and socio-technical realms.

Samenvatting

Motivatie

Van smartphones tot luidsprekers en horloges, Edge AI wordt ingezet op miljarden apparaten om grote hoeveelheden persoonlijke gegevens efficiënt, vertrouwelijk en realtime te verwerken. Met een eenvoudige set trefwoorden, zoals “roep om hulp ”, kan een Edge AI-systeem belangrijke diensten, zoals noodhulp, met een handige spraakopdracht activeren. Hoewel toepassingen van Edge AI veelbelovend zijn, laten veel recente incidenten van vooroordelen in artificiële intelligentie (AI) systemen zien dat Edge AI ook kan discrimineren tussen groepen mensen op basis van geslacht, etniciteit, leeftijd, accent, nationaliteit en andere persoonlijke kenmerken. Met name doordat de fysieke beperkingen en de complexiteit van de ongelijkmatige en gedecentraliseerde werkomgeving van Edge AI om afwegingen vragen bij de inzet van de technologie. Deze dissertatie adresseert de maatschappelijke vraag naar betrouwbare AI, de neiging van vooroordelen in AI-systemen, en de noodzaak om vooringenomenheid in diverse Edge AI-toepassingen te detecteren en mitigeren. Hiervoor zijn generaliseerbare benaderingen vereist die vooroordelen tijdens de ontwikkeling van Edge AI te detecteren en mitigeren.

Doelstelling

Het doel van deze dissertatie is om *design patterns* te ontwikkelen voor het detecteren en mitigeren van vooroordelen tijdens de ontwikkeling van Edge AI-systemen. *Design patterns* zijn een generaliseerbare benadering om gevestigde routines voor het detecteren en mitigeren van vooroordelen in machine learning (ML) vast te leggen. Onderzoekers en ontwikkelaars die Edge AI ontwikkelen, die beperkte ervaring hebben met het detecteren en mitigeren van vooroordelen, krijgen op deze manier gemakkelijk toegang tot deze kennis. Bovendien zouden de *patterns* een instrument moeten bieden voor onderzoekers en ontwikkelaars om nieuwe kennis en inzichten over vooroordelen in Edge AI te delen tussen toepassingsdomeinen.

Overzicht en resultaten

Hoofdstuk 1 motiveert de behoefte vooroordelen in door Edge AI ondersteunde slimme systemen te detecteren en mitigeren. Het hoofdstuk presenteert de wetenschappelijke kennislacunes die dit onderzoek tracht aan te pakken, en schetst de onderzoeksdoelen en -vragen. Verder worden de onderzoeksbenadering en een overzicht van de dissertatie gepresenteerd.

Hoofdstuk 2 schetst de context van dit proefschrift door technische achtergrondkennis over Edge AI te geven. Dit is een rekenparadigma waarin de verwerking van persoonlijke en gevoelige sensorgegevens door ML met behulp van edge computing (deels) plaatsvindt in de periferie van Internet of Things. Verder worden on-device ML, een categorie van Edge AI, en het toepassingsgebied van spraakgestuurde Edge AI in detail beschreven.

Hoofdstuk 3 gaat verder dan de technische fundamenten van Edge AI en bespreekt gerelateerd werk over betrouwbaarheid, rechtvaardigheid en vooroordelen in ML. Het hoofdstuk benadrukt dat de detectie en mitigatie van vooroordelen in de ontwikkeling van Edge AI onderbelicht zijn in de literatuur. Het hoofdstuk beoordeelt ook de huidige benaderingen voor het detecteren en mitigeren van vooroordelen in ML en benoemt uitdagingen in implementatie en kennisoverdrachts die de acceptatie van deze benaderingen in Edge AI hinderen. Het hoofdstuk sluit af met een overzicht van *design patterns* die deze uitdagingen kunnen overwinnen.

Hoofdstuk 4 presenteert design *patterns* om vooroordelen in ML te detecteren en te mitigeren op basis van gevestigde kennis en praktijken. Het hoofdstuk stelt een conceptueel model voor om het ontdekken van *patterns* te begeleiden, identificeert 106 pattern instanties in softwaretools, voegt ze samen tot 23 *patterns* en stelt de *patterns* op in een catalogus. Ten slotte formuleert het hoofdstuk twee recepten die een verzameling *patterns* gebruiken om *Bias Measurement* en *Benchmark Dataset* processen te begeleiden.

Hoofdstuk 5 valideert het nut van de voorgestelde *patterns* en recepten in een ML gebruiksscenario dat sprekerverificatiesystemen onderzoekt. Het hoofdstuk valideert het nut van de *Benchmark Dataset* en *Bias Measurement* recepten voor het detecteren van vooroordelen in vooraf getrainde modellen en bestaande evaluatiedatasets. Vervolgens worden valkuilen in evaluaties van vooroordelen geïdentificeerd die kunnen worden verminderd door het gebruik van *patterns* en recepten. Het hoofdstuk introduceert twee nieuwe *patterns* die zijn ontdekt in het gebruiksscenario.

Hoofdstuk 6 breidt de *patterns* uit en past ze aan op een Edge AI-omgeving. Het hoofdstuk is gebaseerd op de eerste wetenschappelijke studie naar vooroordelen in on-device ML en gebruikt de *patterns* om de impact van modeltraining en ontwerpkeuzes gericht op optimalisatie te onderzoeken in een gebruiksscenario voor trefwoorddetectie. Het hoofdstuk legt de inzichten uit de evaluatie van vooroordelen vast als nieuwe *patterns*, waardoor de patrooncatalogus wordt aangepast op het detecteren en mitigeren van vooroordelen in on-device ML.

Hoofdstuk 7 biedt een overzicht van de conclusies van het onderzoek. Het hoofdstuk

bespreekt de bevindingen, benadrukt de onderzoeksbijdragen, reflecteert op de beperkingen van het onderzoek en doet aanbevelingen voor toekomstig werk.

Hoofdstuk 8 reflecteert op de relevantie van dit proefschrift naast de directe bijdrage gezien de ontwikkelingen op dit moment. Het hoofdstuk pleit voor een ontwerpperspectief op risico's en schade door AI, benadrukt de behoefte aan een verhoogde ontwerp- en systeembenadering in onderzoek naar betrouwbare AI, en roept uiteindelijk op tot onderzoek naar betrouwbare AI dat de systeemgrenzen van AI uitbreidt van technisch naar praktisch en sociaal-technisch gebied.

Preface

A day like today seems like a good day to write the preface of a PhD thesis. It's a good day because the end is so near, but not quite there yet. That makes it a hopeful day. I can write with hope of what is to come, which feels tantalising. Dreaming about the future, being hopeful, is such a quintessential human trait. Also, after three years of waiting, today I was finally rewarded with an exquisite experience that I have patiently pursued: a visit to the Japanese Garden at Clingendael Estate in Den Haag. A small enclave of mossy trees, silent water, and pink blossoms unashamedly showing off their splendour. Since my arrival in the Netherlands this has been the Promised Garden. A tiny patch of land which I passed weekly when going jogging. Summer, winter, autumn, spring. Fenced off from the rest of the estate, a small sign outside its perimeter unseemingly advertising its sparse opening hours to curious passersby.

The Japanese Garden was never open. Like all of us, it was subjected to senseless suppression during the coronavirus pandemic. But today the long wait is finally over. Indeed, the pandemic lies so far in the past that I almost forgot it existed. Strange, when I consider that over half the work in this thesis was conducted in isolation. If there is one thing I have learnt through this, it is that research should not be an isolated endeavour. The goal of science is to increase human knowledge. Implicitly, this refers to *collective* human knowledge. With my PhD drawing to the end, I am acutely aware of how little I know. In the face of individual ignorance, increasing my knowledge would thus be an easy undertaking. Increasing collective knowledge, on the other hand, is hard. Collective knowledge is situated. It exists in relation to others and requires a sense of belonging to a community. Increasing knowledge by contributing to a community makes scientific research a fundamentally collective undertaking. Human knowledge increases if *we* learn. And this cannot happen alone.

Back at the Japanese Garden, I let time stand still for a moment. Visiting the garden started as a quest of desire. Over time my desire to visit the garden turned into hope that it would one day be open. I kept the hope alive by patiently waiting, putting in persistent weekly effort to run past it, catching glimpses of what may lie beyond its boundary. Desire, hope, patience, persistence and ultimately being rewarded with beauty. As the seasons turn once again and spring is in the air, I am surprised to observe my PhD journey reflected in this natural jewel of the Netherlands. Us scientists may not like to cast our profession as a hopeful one, but submitting yourself to a PhD and conducting research is in its essence a hopeful endeavour. We hope to discover, we hope to learn, we hope to find meaning and to do something that matters. So much desire, and so much hope. So much demand for patience and persistence that on a monthly, weekly,

daily basis we, as humble PhD candidates, are constantly confronted with the reality of our vast lack: lack of knowledge, lack of publications, lack of citations, collaborations, presentations.

A lot appears to have changed since September 2019 when I started my PhD. Of course there was the pandemic where everything changed, while nothing changed. Daily monotony. Except for the radical shifts happening in the space of technology. Artificial Intelligence, repeated so many times on the radio that now even my mother can make sense of my research topic. Things change. Now, so close to the end, I am proud not to have let the"lack of" get me down. Creating new knowledge requires patience, persistence and a good deal of luck, only to seem obvious in retrospect. May we continue to honour the pioneering effort that it takes to do scientific research. May we continue to demand patience and persistence from ourselves to ensure that our work serves society. I choose to stay hopeful. Today, my hope is that this thesis contributes a small sliver of actionable insights to our expansive, ever-changing world, in which designing Artificial Intelligence responsibly and reflectively, to be inclusive, inspire harmony and promote human flourishing, is as relevant as ever.

Wiebke (Toussaint) Hutiri
6 May 2023

Japanese Garden, Clingendael Estate, Den Haag. *Photo taken 6 May 2023.*

Acknowledgements

It takes a village to raise a child. It takes a community to complete a PhD.

To **Aaron Ding**, you have fuelled my confidence and encouraged me to aim high, you have guided me while giving me freedom to explore and be curious. You have supported my sometimes unconventional decisions and have encouraged me to expand my intellectual horizons and networks. Thank you for taking on this journey with me. Your committed supervision, your consistent presence and your unfaltering faith in my capabilities have helped me fortify my scientific roots and succeed in completing a PhD that I am proud of. Thank you. **Marijn Janssen**, you were the metronome that kept me in check. Your experience and oversight have given me a deep appreciation for the role of a promoter on the PhD journey. I fully trusted your guidance and always welcomed your frank input, probing questions about concept clarity, constant reminders about academic integrity, and your support of my independence. It felt like you had access to a meta-map for the path I was walking, and you could share the directions, without knowing my end destination. My heartfelt gratitude.

Ellen Schwencke-Karlas, **Minaksie Ramsoekh**, **Laura Bruns**, **Diones Supriana**, **Olivie Beek**, **Christine Bell** and **Jolien Ubacht** - over the years you have supported me in so many ways, on so many tasks: from finding time for meetings in the always busy schedules of professors, to tracking down lost parcels, arranging my contract to accommodate work placements or remote work in South Africa, helping me replace a stolen laptop, and so much more. Thank you! Without you, navigating the day-to-day practicalities of being a PhD candidate would have been intractable.

To the brilliant professors and researchers at TBM-ESS-ICT, with whom I've had the pleasure to exchange ideas and engage in debate: your research, input and presence is what makes our section exciting and an attractive place for PhDs like myself to come and study. Thank you, **Nitesh Bharosa**, **Roel Dobbe**, **Nadia Metoui**, **Jacobien Oosterhoff**, **Mark de Reuver**, **Boriana Rukanova**, **Yao-Hua Tan**, **Jolien Ubacht** and **Anneke Zuiderwijk-van Eijk**. To **Casper Chorus**, you may have moved on from TBM, but your leadership during the coronavirus pandemic was an inspiration to me and made a big difference to how I experienced some of the hardest months of my PhD.

I am grateful to have walked this research journey alongside others with whom I have been able to engage in productive collaborations and from whom I have received mentorship and guidance. **Akhil Mathur** and **Fahim Kawsar**, the time at Nokia Bell Labs in Cambridge in 2021 stands out to me as a turning point in my PhD. After more than a year of working in isolation from home, being welcomed at your offices, challenged and supported in pursuing exciting research, set the tone for the remainder of

my PhD. Thank you for your generosity. **Odette Scharenborg** and **Tanvina Patel**, it has been such a joy to discover that colleagues at TU Delft were working as passionately on detecting and mitigating bias in speech technologies as myself. I have learnt a lot from both of you, and thank you for your openness towards embracing our interdisciplinary collaboration. **Olya Kudina**, **Jered Vroon**, **Jacky Bourgeois**, **Julian Harty**, **Gürkan Solmaz** and **Ella Peltonen**, I fondly look back at our creative journey of discovering data daemons through design futures. **Alejandra Gómez Ortega**, from data daemons to virtual and voice assistants, in you I have found a consistent match for my curious and creative research and running ambitions.

Sem Nouws, **Antragama Abbas** and **Rijk Mercuur**: what would a PhD be without office mates? You have accompanied me, challenged me, been my sounding board and the witnesses of the daily questions, frustrations and celebrations, both professional and personal, that doing a PhD brings with it. Without you, my PhD would have been much less fun and much less meaningful. **Nina Bohm**, I met you in my first week at TU Delft and we decided to be friends. How lucky for me, and what a great choice by us! **Annika Herth**, I don't remember how we met, but I think it was inevitable. To the chats, and chills and of course four packets of corona pancakes with **Roman Henning** and **Vladimir Sobota** mit etwas deutschem Geschwätz.

My lab mates, **Dewant Katare**, **Prachi Bagave**, **Reni Sulastri**, **Marcus Westberg**, thank you for giving me honest and critical feedback during lab meets and for sharing your time at TU Delft with me. Marcus, you have made me laugh so many times, thank you for bringing honesty and good, dry humour to lunches. **Íñigo Martinez de Rituerto de Troya**, the third Observer, and all the amazing humans that have frequented the third floor: **Fernando Kleiman**, **Wirawan Agahari (Aga)**, **Ini Kong**, **Antonia Sattleger**, **Eva De Winkel**, **Mannat Kaur**, **Svenja Bielefeld** and **Christine Milchram** – you are a big reason of why I have loved being on campus, and why I continue to return. **Jaewon Choi**, **Lakshmi (Manasa) Kalanadhabhatta** and **Jing Yang** I have so many good memories of spending the summer of 2021 at Nokia Bell Labs in Cambridge with you. **Jacob Foster**, **Erika Cartmill**, **Kensy Cooperrider** and **Amanda McAlpin** and the Diverse Intelligences Summer Institute (DISI), the summer of 2022 in St Andrews was a mind-expanding journey into the mysteries of the many intelligences roaming this planet. Thank you for bringing us together. All the **DISI fellows**, I've learnt so much from all of you and will refrain from listing you individually, because you're many. **Jenny Liu Zhang**, without knowing it, you brought design patterns back into my life and gifted me the narrative arch of this thesis. Thank you!

To the FairEVA team, **Anna Leschanowsky**, **Carolyn Quinlan**, **Casandra Rusti**, **Lauriane Gorce** and **Michaela Pnacek**, working with you and harnessing our very diverse, collective genius to make a difference in the world kept me sane and was a highlight of my last four years. We were a power team, and I've learnt a lot from each of you. Thank you for trusting me and following me on a journey into the unknown. **Temi Popo**, **Mehan Jayasuriya** and the **Mozilla Foundation**, thank you for being will-

ing to take risks and live the values you stand for. You've enabled me to compliment my research on trustworthy AI with practice and activism, and I am truely grateful for that. **Borhane Blili-Hamelin** and **Bernease Herman**, co-creating The Zen of ML with you was a lot of fun. In many ways this thesis feels like an evolution of what we started. Thank you for your imagination and collaboration. **Kathleen Siminyu**, **Bogdana Rakova** and **Kathy Reid**, your work has inspired me from afar. Our journeys are young and just beginning, I look forward to finding opportunities for our work to connect in future.

Sara Hooker, your refreshingly nonchalant brilliance inspires me. Thank you for having a generous soul, for being passionately committed to increasing access to AI and for opening doors for me. **Daniel Situnayake**, thank you for being as excited about the responsible design of Edge AI as I am, for trusting me to review your book, and for being so intentional about making the circle bigger and including me. **Chris Fourie**, **Jade Abbot** and **Pelonomi Moiloa**, thank you for being new friends, compatriots, my sources of motivation and welcome distraction during some of the long hours of writing up my PhD thesis in Johannesburg. **Deshen Moodley** and **Tommie Meyer**, you have given me the gift of a joyful first research experience during my Masters. Thank you for instilling many of the values, habits and skills that have allowed me to succeed in my PhD.

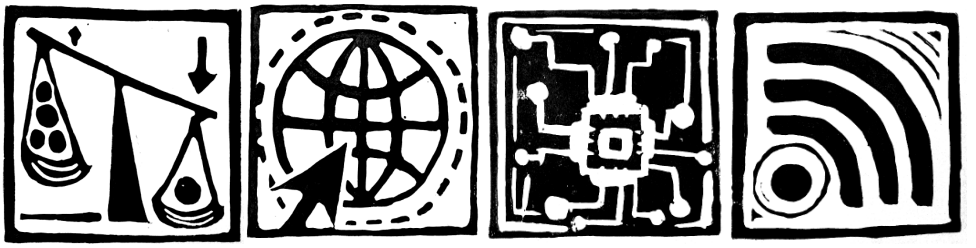
Matt Davey and **Margot Leger**, you gently nudged me to come to the Netherlands and became my family away from home. I'm glad that I will find you back in Cape Town. **Familie Swanepoel**, thank you for being my South African refuge in Amsterdam, always ready to fire up the Sunday braai. **Mama**, du hast mir seit meiner Kindheit die Freude am Lernen und Entdecken ermöglicht. Danke, dass du mich auf deinen Schultern hast stehen lassen, dass du mir, ohne zu zögern, die Tür in eine Welt geöffnet hast die dir unbekannt und manchmal vielleicht auch ungeheuerlich war. Danke, dass du mich als mutige und starke Frau erzogen hast, die der Welt gewachsen ist. Ich bin wer ich bin durch dich. **Papa** and **Jenni**, bei euch haben wir so viele friedliche Nachmittage und Abende verbracht. Papa, ich danke dir für deine Aufrichtigkeit, für deine Bereitschaft dich, trotz Sturheit, doch manchmal von deiner Tochter beeinflussen zu lassen, für deinen Optimismus und dein Vertrauen in mich. Jenni, I thank you for being a light that radiates so much love in this world. We feel at home when we come to visit you. **Heike** und **Birte**, Gespräche und Gedankenaustausche mit euch über mein Leben hinweg haben meine Gedanken geprägt und beinflusst. Ihr inspiriert mich, und ich bin stolz eure Schwester zu sein.

Neo Hutiri, my beloved husband, who would have thought that we can incubate our marriage at 9000km distance? You are my other half and my complement. You inspire and motivate me. I'm ever so grateful that patience is one of your greatest strengths. Now that the PhD is done, let's do life together!

This thesis was written without the assistance of a large language model - not out of principle, but because I love to write. Don't use AI to replace you in doing work you love.

1

Introduction



1

1.1. Edge AI in Smart Systems

For decades, the vision of technology-enabled smart systems has captured the fascination of scientists, technologists, futurists and governments. Smart systems promise that granular, high-resolution data can enable personalised services [305] and industrial efficiency [338], improve citizen well-being [330] and support evidence-based decision-making [293]. The idea of smart systems is no longer merely a vision. Every year, millions of sensors, like microphones and cameras, are integrated into the Internet of Things (IoT) to intimately and continuously connect people and objects [132]. In 2018 an estimated 17.8 billion connected devices were reported in use [163]. This number is projected to almost double by 2025. From energy [96, 314, 333], water [315] and transport [169, 243] systems, to agriculture [74] and health [81], buildings [202], homes [250] and even the human body [89], there is hardly an area of public and private life that has not been reimagined as a producer and consumer of vast quantities of sensor data.

Once collected, sensor data becomes valuable when it is processed to detect events, to activate and control systems, or to be analysed and aggregated to reveal individual or societal behaviours. For example, security breeches such as electricity theft [206], health incidents like falls [50] and other anomalous events [233] can be discovered from patterns in sensor data. In the electricity and water sectors, smart meter data can be clustered to profile customers based on their consumption behaviour [73, 288]. Wearable and mobile data can be used to recognise what physical activities a person is doing [311]. Autonomous vehicles rely on camera sensor data to perceive their environment [181], and voice assistants collect speech data with microphones to receive instructions and perform actions to entertain, enlighten or sometimes infuriate us [260].

With the aid of artificial intelligence (AI) and specifically machine learning (ML), the large volumes of data created by sensors in the IoT can be turned into real-time predictions and automated decisions that enable new services in the public and private sectors [26, 255]. However, to augment the IoT with AI, large volumes of data need to be transferred to remote cloud servers for processing. Yet sending sensitive and personal data, such as human speech, to the cloud raises the risk of exposing private information [332]. Potential benefits of AI-augmented, cloud-based smart systems thus stand in tension with a world in which pervasive monitoring and surveillance systems infringe on citizens' personal privacy and autonomy [337].

Alternatives to conventional cloud-based data processing are necessary if emerging smart systems are to observe the right of citizens to privacy[1]. Edge AI, as depicted in Figure 1.1, is a computing paradigm that shifts ML data processing from the cloud to localised servers and devices at the edge of the communication network, closer to the data sources in the IoT [336]. This has important benefits: by performing computa-

[1] In the EU, citizens' right to privacy is set out in Article 8 of the Charter of Fundamental Rights.

tions locally and thus restricting data transfer, Edge AI can draw insights from data in situ, safe-guarding data privacy and ensuring reliable operation, independent of communication network delays or interruptions [64].

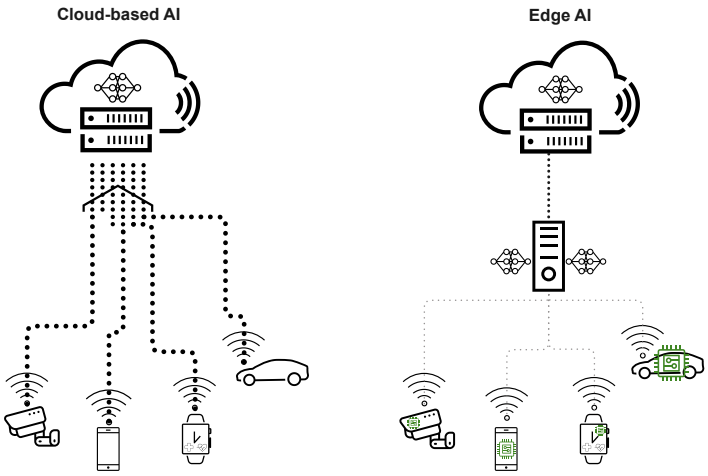


Figure 1.1: Cloud-based and Edge AI. Cloud-based AI transmits data to centralised, remote servers for processing. Edge AI processes data in-situ, on or close to the device that collected it.

1.2. Motivation for Detecting and Mitigating Bias in Edge AI

With smart systems and Edge AI as backdrop, this thesis is motivated by the societal demand for trustworthy AI, by evidence of bias in AI systems, and by the susceptibility of Edge AI systems to bias. This section discusses these three motivating factors.

Trustworthy AI

Five years ago, a plethora of failures in algorithmic systems highlighted the lack of fairness, accountability and transparency in the development of AI technology [321]. This deficit, alongside concerns of surveillance and a lack of oversight mechanisms and regulations, called into question whether AI technology ought to be trusted. Trust presupposes that someone is willing to take an action, contingent on another party, under the belief that they will not be betrayed due to ill intentions or lack of competence of the other party [167]. Trust is warranted if the other party is deemed to possess attributes that make them worthy of being trusted. While trust is a judgement, trustworthiness refers to attributes that justify trust. From Cambridge Analytica's ill-intentioned personal data breeches that facilitated political interference in the UK and US [213], to

1

Facebook's (now Meta's) incompetence at monitoring and preventing its platform and algorithmic recommendation engine from being used to coordinate a genocide against a Muslim minority in Myanmar [225], ample evidence existed that AI systems were developed in a manner and under conditions that did not warrant trust. Since these disasters, mistrust towards the development of AI technology has continued. Today, the rapid and uncontrolled dissemination of new AI tools for text, image, video and voice generation has escalated calls to interrogate the trustworthiness of AI technology [57].

Trustworthy AI emerged as an umbrella term to express attributes that AI technology ought to have in order to meet ethical principles that would make the technology worthy of human and societal trust [119]. The demand for trustworthiness in the development of AI is not unique, as trust and trustworthiness have been studied and formalised previously in many technology domains, including the IoT [190, 327]. However, in contrast to IoT and cyber-physical systems that consider trustworthiness in light of technical attributes of robustness, reliability, safety, security and privacy [48], the conceptualisation of *trustworthy AI* also comprises societal and institutional requirements, like transparency, accountability, non-discrimination and fairness [88]. Fairness and non-discrimination in AI have received particular attention in the academic literature [12], as the reliance of ML on vast quantities of data makes it prone to bias, a skew in predictive outcomes that favour some groups of people while disadvantaging others [16]. At the inception of this thesis, calls were emerging for trustworthy Edge AI [62], yet bias, fairness and discrimination had not been studied in this setting.

Bias in AI Systems

Bias, in general terms, refers to a skewed or slanted perspective [90]. When humans make decisions, emotional and social factors influence preferences and choices [140]. This "skews" individual perspectives, and makes cognitive bias a characteristic feature of how humans interact with the world. When decisions lead to actions, cognitive biases do not remain mental constructs, but structure and organise the world that is shaped through human actions. Consequently, cognitive biases find their way back into the artificial world designed by humans [100], where they become embedded as technology bias. Biased technology carries significant social consequences when it automates decisions or leads to actions that systematically assign undesirable outcomes or deny opportunities to individuals or groups without justification [90]. Colloquially bias is often used to refer to instances of prejudice [66]. In the literature, bias is viewed as a source of unfairness in ML systems [192].

In the past years there have been many reports of biased AI systems that systematically discriminated against groups of people based on their personal attributes. For example IBM, Microsoft and Face++'s commercial facial analysis software systems were significantly more likely to misclassify darker-skinned females [29], Amazon's internal human resources recruitment tool disqualified applications that included the word

"women's" [49] and Apple offered women only a tenth of the loan amount that it offered their male partners when the company first launched their credit card [219].

In the literature many studies have revealed evidence of bias in ML applications [25, 240, 325]. Bias can originate at various stages in the end-to-end ML workflow, for example when datasets are collected and curated, during model training or evaluation, when conceptualising benchmarks or due to other design choices [277]. In contrast to other technologies, biased computer systems are a particular concern, as they can amplify and spread bias cheaply, at a scale that far exceeds that of individual humans [90]. In ML systems the tendency for bias is heightened due to their reliance on and feedback loops created by data [226]. Researchers have been studying approaches to mitigate bias across a wide range of ML applications [230] to meet the growing societal [245] and regulatory [78, 284] pressure for inclusive, fair and non-discriminatory AI systems.

Anticipating Bias in Edge AI

Edge AI applications are prolific in people's daily lives and their adoption is rapidly expanding. For example, by 2024 an estimated 8.4 billion voice assistants, a number roughly equal to the human population, is predicted to be in use [84] and will deploy voice-activated Edge AI to launch IoT services. Like cloud-based AI, Edge AI systems use ML techniques that are developed with and operated on data. The reliance of ML on data gives strong reasons to anticipate bias in Edge AI. At the same time, Edge AI has unique attributes that make it difficult to detect bias, and that may even amplify bias. In Edge AI, sensor data, hardware platforms and software implementations are heterogeneous and decentralised [36]. Moreover, many edge servers and devices are resource constrained, with limited computing power, memory and data storage. Devices are frequently battery operated, and thus have limited energy resources.

These physical restrictions, together with the complexity of heterogeneous and decentralised operating environments, make the deployment of AI to the edge technically difficult and require Edge AI developers to make design trade-offs to balance the predictive performance of ML models with hardware and energy consumption constraints [55]. This thesis speculates that design trade-offs made during Edge AI development may not only impact predictive performance and hardware resources, but also bias. As Edge AI is increasingly integrated into transport and health care systems, homes, schools and public spaces [228], unobserved bias in Edge AI poses a risk of large scale exclusion and discrimination in the physical world. This makes it important to detect and mitigate bias in the development of Edge AI systems.

This thesis postulates that Edge AI practitioners have a responsibility to detect and mitigate bias during the development of Edge AI. Practitioners are considered to be people who work in the technical realm, where they can occupy many different roles (e.g. researchers, developers or product managers), in any stage of the Edge AI or ML development workflow. Practitioners may be tasked to detect and mitigate bias in Edge AI development, or they directly contribute to the development of Edge AI products or

services and may wish to detect and mitigate bias out of their own volition. Practitioners may or may not code in any programming language, and they are likely to interface with non-technical stakeholders either directly in their team, or in their organisation. Typically Edge AI practitioners will have no prior experience in detecting or mitigating bias in ML.

1.3. Scientific Gap

Bias detection and mitigation in ML are an active area of research, yet no prior studies have investigated bias in Edge AI. This section presents two perspectives that contribute to the scientific gap that this thesis investigates. Firstly, there is a gap in research on how the unique constraints of Edge AI impact bias. Secondly, while studying these unique constraints could be facilitated by adapting best practices on bias detection and mitigation from ML, no prior research has studied approaches to transfer relevant knowledge on bias from ML to new domains. Based on these gaps, the section positions design patterns as a prospective solution to transfer knowledge on bias detection and mitigation from ML to Edge AI, and presents their development as the scientific gap explored in this thesis.

Limited Research on How Edge AI's Unique Constraints Impact Bias

Prior research on bias in ML has been driven by studies on fairness, accountability and transparency in algorithmic decision-making systems [1, 192, 230]. Bias and fairness have also been widely studied in computer vision [29, 137, 316, 325], recommendation systems [37, 182], natural language processing [24, 25, 173] and automatic speech processing [155, 281]. These ML application areas are also relevant to the Edge AI domain, where research has focused on deploying systems for computer vision [32, 185, 275], speech processing [170, 312, 334] and other sensor data modalities [144, 224, 242] to edge and end devices. On the one hand Edge AI systems are similar to those studied in prior research on bias in ML: they use data and algorithms to train models that are deployed for perception and to automate decision-making processes. On the other hand Edge AI is also distinctly different.

Edge AI applications are pervasive, distributed, hardware-constrained, localised, context specific and tightly integrated into the physical world [36]. These factors play a role in the design of Edge AI applications. For example, limited memory and compute require ML models to be compressed in size before they can be deployed to a device [296], or power constraints require energy efficiency to be considered alongside predictive performance during model training [203]. However, current research on bias in ML abstracts the hardware and energy resources involved in ML training and inference, under the assumption that systems are trained and deployed on the cloud where energy and computing resources are abundant and homogeneous. While many approaches and practices from ML will be relevant for detecting and mitigat-

ing bias in Edge AI, the distributed, hardware-dependent, heterogeneous and context-specific nature of Edge AI presents unique challenges that may amplify bias and that have not been studied in the literature. Bias in Edge AI thus needs to be studied in its own right. To date, research on bias in Edge AI and approaches for detecting and mitigating bias in Edge AI development are very limited and remain an open research problem [26, 63, 143].

Lack of Transferable Knowledge on Bias Detection and Mitigation in ML

Researchers who aim to study bias in ML but who have limited prior knowledge in the domain encounter challenges when engaging with the nuanced approaches of this interdisciplinary research area. For example, in a well known talk Narayanan [210] disambiguated fairness definitions and their meanings with the aim of encouraging technical researchers to embrace the reality that fairness has no single definition, and that "technical considerations cannot adjudicate moral debates" [211]. This insight is not obvious to many people. In my personal experience I observed that practitioners who are new to bias detection and mitigation oftentimes neglect to consider the variability of outcomes that different fairness definitions entail. Quite contrary to Narayanan's intent, there are others who continue to view the variety of fairness definitions as a weakness that makes fairness intractable, and those that focus on creating new fairness metrics to measure bias, without considering the underlying outcomes that the metrics promote.

Even if practitioners appreciate these distinctions, they often experience difficulties navigating and prioritising algorithmic fairness techniques, which are nuanced and context specific [56, 122, 166]. As a consequence, practitioners who chart out trajectories for detecting and mitigating bias in new domains can invest significant effort into rediscovering solutions that have already been addressed, repeat mistakes or use approaches that are contested in the literature. For example, Deng et al. [56] note that developers who are unfamiliar with work in ML Fairness frequently attempt to mitigate bias with a "fairness through unawareness" approach, which is likely to increase, rather than reduce bias. If established practices and state of the art knowledge on bias detection and mitigation in ML were readily accessible, and if this knowledge was transferable between Edge AI applications, detecting and mitigating bias in Edge AI would be greatly facilitated.

Design Patterns as a Prospective Solution for Knowledge Transfer

Approaches for transferring established knowledge on bias detection and mitigation from ML to other disciplines and between application domains have not been studied in the literature. However, the problem of transferring knowledge about design artifacts and processes is not unique to bias in ML, and solutions have been developed in other domains to overcome this challenge. Design patterns present an approach to knowledge transfer that is well established in object-oriented programming and soft-

ware engineering [93, 184, 231, 246], information systems [302, 309], architecture [3] and other engineering design disciplines [9, 99, 205].

In software engineering, design patterns capture and convey design ideas as templates that are useful to developers and reuseable across several contexts and applications. The generalisable form of design patterns makes them effective for capturing and communicating design knowledge so that it can be reused in future projects. Gamma et al. [93], who pioneered the development of design patterns for object oriented programming, defined a design pattern as "a solution to a problem in a context". Design patterns present a promising approach for capturing established knowledge and practices on bias detection and mitigation in ML, so that this knowledge can guide bias detection and mitigation during the development of Edge AI systems. While some recent works have explored design patterns for building responsible and trustworthy AI [71, 75, 177, 276], they consider patterns on a high level, or do not focus on bias detection and mitigation specifically. The study and development of design patterns for detecting and mitigating bias in Edge AI thus remains a gap in the literature and is a novel undertaking worthy of scientific enquiry.

1.4. Research Aim

This thesis aims to develop design patterns for detecting and mitigating bias in the development of Edge AI systems. The objective for creating design patterns is to explicitly capture established knowledge and practices on bias detection and mitigation from ML, so that this experience can be reused by practitioners in Edge AI. The stated aim can be divided into three further goals. The first goal is to identify design patterns that capture current approaches for bias detection and mitigation in ML. The second goal is to demonstrate the validity of the design patterns for detecting and mitigating bias in general ML applications. The third goal is to extend the design patterns to the specific context of Edge AI, and to adapt them if needed. The goals and corresponding research questions are summarised below:

Goal 1:

Identify and capture relevant knowledge and practices for bias detection and mitigation from ML as design patterns.

Research Question 1 (RQ1): *Which established approaches for bias detection and mitigation in ML can help practitioners detect and mitigate bias in new domains?*

Goal 2:

Validate the utility of the proposed design patterns in a ML use case where bias has not been studied previously.

Research Question 2 (RQ2): *To what extent are the design patterns elicited in RQ1 useful for detecting and mitigating bias in ML in new domains?*

Goal 3:

Extend the design patterns to an Edge AI use case, adapting them to incorporate new knowledge specific to detecting or mitigating bias in Edge AI systems.

Research Question 3 (RQ3): *Which aspects of Edge AI may affect bias?*

The patterns should be as comprehensive as possible. However, as bias detection and mitigation in Edge AI are nascent, new knowledge will continue to emerge and the patterns are expected to evolve over time. The patterns are thus not expected to be complete. Instead, they should be extensible so that they can be revisited and revised as research and practice evolve.

1.4.1. Scope

As noted in Section 1.2 and discussed in more detail in Chapter 2, Edge AI is an enabling technology with applicability to many sectors and applications. Even if bias ought to be investigated across applications, validating and extending the design patterns to every type of application is not feasible within the scope of this thesis.

Voice-Activated Edge AI

The thesis scope is thus constrained to voice-activated Edge AI. This scope was chosen for a number of reasons. Voice-activated Edge AI already exists in abundance in voice assistants and voice interfaces on mobile phones and smart speakers. These applications process human voice data, which contains a wide spectrum of personal information [271]. Voice data is thus particularly privacy sensitive, which makes the privacy-preserving attributes of Edge AI desirable. Voice-activated Edge AI also serves many purposes and can be used to invoke numerous services – from asking a smart speaker at home to play music, to enabling route finding in a car, calling an elevator or even calling for help in an emergency situation. While different uses carry different risks when a voice-activated service fails to be invoked correctly, in all these situations humans are directly affected. This makes potential bias a matter of concern, and juxtaposes the trustworthy AI aspect of privacy against non-discrimination, fairness and inclusion. Even though the use cases studied in this thesis are focused, the patterns are expected to be general and applicable beyond voice-activated Edge AI, in particular to applications that process human-related and personal data.

Technology Bias Contained in Artifacts

Bias in ML, as will be discussed in Section 3.3.2, is maintained through reinforcing feedback loops that emerge from complex interactions in socio-technical systems. Amongst these feedback loops and interactions, this thesis focuses on technology bias contained in artifacts, namely the data and processing components in end-to-end ML development workflows of Edge AI applications. It does not study bias in humans that create and interact with Edge AI, or in institutions that enable Edge AI to be created and

maintained. This view presents a partial perspective of bias in Edge AI, as technological components that lead to bias do not arise in a vacuum, but from individual choices, social values and institutional processes. However, focusing on technology bias draws a boundary that is necessary to make tangible progress towards the research objective. In Chapter 8 I will argue that the realisation of trustworthy AI requires this boundary to be extended, and that Edge AI development should be considered from a broader socio-technical perspective.

1.5. Research Approach

This thesis studies design patterns for detecting and mitigating bias in Edge AI by conceptualising, designing and validating them. The research follows a design science research approach with emphasis on artifact development. This kind of research promotes the practice of "design as research" [117] and is premised on the idea that practicing design and developing artifacts can result in clear knowledge contributions.

Design as Research

Simon [268] distinguished the natural world from the artificial world created by humans, and positioned design as a central process for creating artificial objects, or artifacts. Artifacts are designed to have desired properties that support humans in attaining particular objectives. Design is thus concerned "not with the necessary but with the contingent – not with how things are but with how they might be" [268, p. xx]. Simon introduced design science as an area of research that studies the artificial world created by humans, complementary to the natural sciences. Design science is an appropriate perspective to apply to the study of ML and Edge AI systems, which are, after all, artifacts created by humans to attain particular goals. Design science research is a mature area of study in disciplines like information systems [117], where research has been categorised in two classes. Firstly, the design of innovative artifacts that contribute new knowledge constitutes research in and of itself. Secondly, researchers can also conduct design science research by studying designed artifacts, designers and design processes. This thesis adopts the former perspective and aims to contribute new knowledge by approaching *design as research*.

Hevner et al. [118] prescribe seven guidelines for conducting this kind of research. First, it must produce a viable artifact. Second, the artifact should present a solution to an important and relevant business problem. Third, the utility, quality and efficacy of the artifact must be rigorously evaluated. Fourth, knowledge gained through artifact design must be clear and verifiable, and can contribute to the artifact itself, to design foundations or methodologies. Fifth, artifact construction and evaluation methods must be conducted rigorously. Research rigour is "derived from the effective use of ... theoretical foundations and research methodologies" [118], for example by applying data collection and empirical analysis techniques in an appropriate manner. Sixth,

design science research should be viewed as a search process, in which incremental and iterative design cycles lead a researcher to a solution of a problem. And finally, design science research must be communicated to technology and management-oriented audiences.

The problem that this thesis addresses, namely detecting and mitigating bias in Edge AI, is primarily motivated by a societal call for trustworthy AI, not by a business problem. While this may affect how the utility and efficacy of the developed artifact are considered, it has no effect on the design science research process.

The Design Science Research Method

The Design Science Research Method (DSRM) of Peffers et al. [227] was found to be most suitable for this research, and was thus adopted. The DSRM consists of six design activities that can be applied in repeated iterations. Regarded linearly, the activities are:

1. Problem identification and motivation
2. Definition of objectives for a solution
3. Design and development
4. Demonstration of the artifact in context
5. Evaluation of the artifact's effectiveness and efficiency
6. Communication of findings

The design science research process itself does not need to be linear, and can start at any activity other than the last where findings are communicated. Iterations typically involve a develop-demonstrate-evaluate cycle that includes activities 3, 4 and 5. Evaluation methods can be observational, analytical, experimental, testing or descriptive [118]. This thesis validates the design patterns through empirical and analytical investigations of bias in use cases, first in ML and then in Edge AI, to demonstrate their merit. It thus refers to develop-demonstrate-validate cycles. The validations examine whether the design patterns serve their intended purpose of bias detection and mitigation. While further quantitative evaluations can be conducted in future research, this lies out of the scope of this thesis.

Design Iterations

Even though skilled researchers can communicate the results of their work in a structured and ordered fashion, the actual research process is often unpredictable. In design science research, where iteration is a fundamental attribute of the research process, this is particularly true. I thus provide a retrospective account of the experimental and exploratory process that guided this thesis. Figure 1.2 visualises how the DSRM has been applied. The research entry points were two-fold, problem-centered and context initiated. On the one hand the research was initiated by the aforementioned problem of detecting and mitigating bias in Edge AI. However, identifying and motivating the problem did not lead to the discovery of design patterns as a solution for solving it.

1

Instead, to overcome the lack of prior work in this area, the research was also initiated by conducting a preliminary study of bias in a suitable context of voice-activated Edge AI in the speaker verification domain. This preliminary study sparked the idea of developing design patterns, and lead to defining objectives of what such a solution could entail.

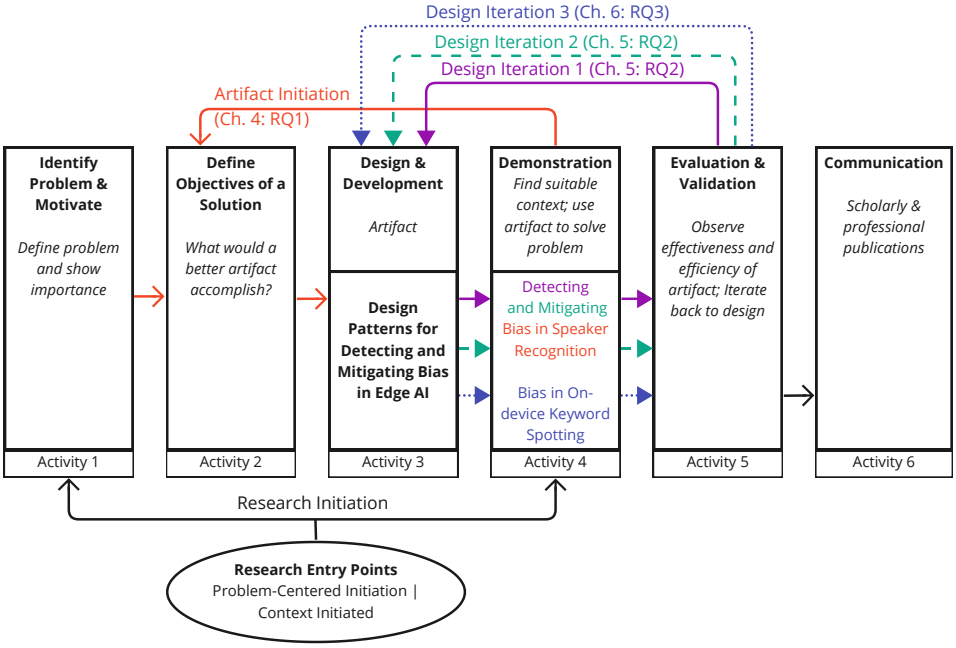


Figure 1.2: The Design Science Research Method of Peffers et al. [227] adapted to this thesis.

Following the research and artifact initiation (orange arrows), I launched the artifact design and development by identifying patterns to detect and mitigate bias in ML from existing software tools. This process addressed RQ1 and accomplished Goal 1. I then returned to the speaker verification use case to demonstrate and validate the utility of the identified design patterns for bias detection (purple arrows). This allowed me to assess the utility of the design patterns for detecting bias in a ML use case, thus responding to the first part of RQ2 and Goal 2. The validation completed the first develop-demonstrate-validate design iteration. The design patterns were updated as needed and I again returned to the speaker verification use case to demonstrate and validate the utility of the patterns for mitigating bias in two further empirical studies (green arrows). With these experiments I completed the second develop-demonstrate-validate cycle, and also validated the design patterns for mitigating bias in a ML use case, thus responding to the second part of RQ2 and Goal 2. In the final develop-demonstrate-validate design iteration (blue arrows) the design patterns were extended and adapted to an on-device keyword spotting use case. This use case

demonstrated the utility of the patterns in an Edge AI setting, and investigated Edge AI specific aspects of bias detection and mitigation. The design iteration thus provided a response to RQ3 and accomplished Goal 3. The patterns were updated one more time before the research was concluded and results were communicated.

Anticipated Knowledge Contribution

Using the design science research knowledge contribution framework of Gregor and Hevner [103], this thesis studies a problem context with low maturity (bias in Edge AI has not been studied), using a solution with high maturity (design patterns have been in use for half a century). The kind of contribution that it aims to make can thus be categorised as exaptation; extending design patterns, a known solution, to a new problem context, bias detection and mitigation in Edge AI. I view the potential contribution of design patterns as nascent design theory, aimed at providing prescriptive knowledge on how to build Edge AI systems that are inclusive, non-discriminatory and fair. At the same time, this thesis also aims to contribute descriptive knowledge by investigating aspects of speaker verification and Edge AI systems that have previously not been studied.

1.6. Thesis Outline

The structure of this thesis is shown in Figure 1.3. Following the Introduction, **Chapter 2** provides technical background information on Edge AI, on-device ML, and voice-activated Edge AI. These technologies underpin the research aim and the use cases that are investigated in later chapters. Readers who are already familiar with these technologies may skip this chapter.

Chapter 3 introduces concepts and theory that underpin the development of the design patterns. The chapter starts by reviewing, comparing and then synthesising trustworthiness aspects from AI and the IoT for Edge AI. It then proceeds to review bias in technology, and related work on detecting and mitigating bias in ML. Lastly, this chapter reviews design patterns and their adoption in software engineering and AI.

Chapter 4 initiates the artifact design and development process. The chapter addresses RQ1 and Goal 1 and identifies established knowledge that is used to detect and mitigate bias in ML in practice. The chapter draws on personal practice and reverse engineers bias evaluation software tools to conceptualise, discover and analyse design patterns for detecting and mitigating bias in ML. It then proposes two collections of patterns as recipes that can be used to guide practitioners when measuring bias or curating benchmark datasets. The main contribution of this chapter are the design patterns, organised in a pattern catalogue, and the pattern recipes.

With the design patterns established, **Chapter 5** completes the first and second design iterations by validating the patterns of the two recipes in a speaker verification use case. The chapter addresses RQ2 and Goal 2 and makes three main contributions.

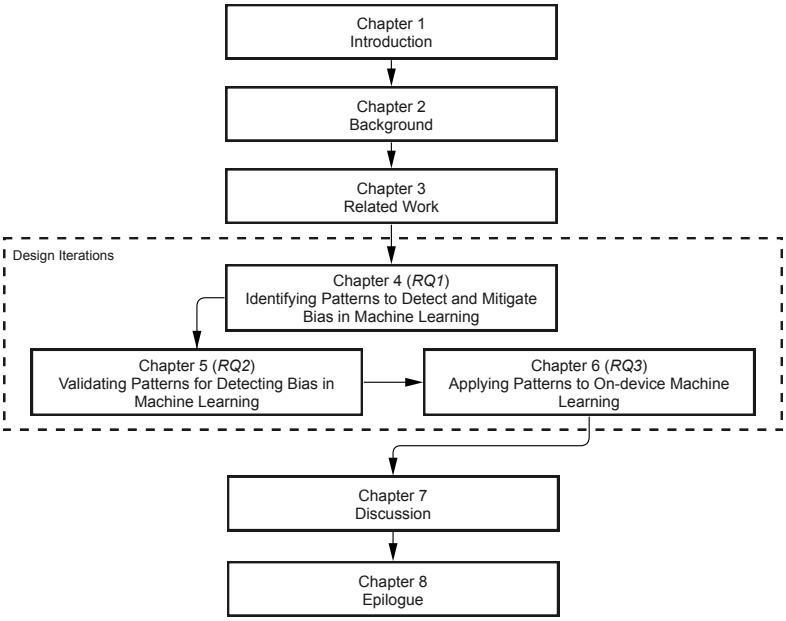


Figure 1.3: Thesis outline

Firstly, it validates the utility of the patterns for detecting bias in a ML use case. Secondly, it presents two experiments in which the recipes are used to demonstrate pitfalls in bias evaluations. The pattern recipes are shown to be useful for overcoming these pitfalls, and the chapter proposes mitigating actions. The third contribution of the chapter is on the application level, as the experiments that have been conducted present new empirical and analytical evidence of bias in speaker verification systems and their development processes.

In **Chapter 6** the pattern recipes are used in a second use case focused on on-device keyword spotting, a common Edge AI application. The chapter addresses RQ3 and Goal 3. This use case demonstrates and validates the utility of the patterns for conducting a reliable bias evaluation in an Edge AI setting. The empirical experiments conducted in the use case, together with the bias evaluation, surface sources of bias in Edge AI that have not been considered in prior research on bias in ML. These insights are then captured as new design patterns to expand the pattern catalogue. The chapter contributes to this thesis by demonstrating the utility of the design patterns in an Edge AI use case, and with new insights on sources of bias in on-device ML that have thus far been unobserved. Chapter 6 completes the iterative development of design patterns for detecting and mitigating bias in Edge AI.

Chapter 7 summarises the findings of the thesis and highlights its scientific and societal contributions. The chapter also discusses the limitations of the design patterns

and their applicability beyond the use cases that have been studied, before making recommendations for future research.

Finally, the Epilogue in **Chapter 8** reflects on the process of conducting this research and the contribution that design patterns may make to the wider community of researchers and practitioners who are working on making AI systems more trustworthy.

2

Technical Background

This chapter presents Edge AI as a computing paradigm for processing personal and sensitive sensor data. First, the constituent technologies that comprise Edge AI, namely the Internet of Things (IoT), edge computing, artificial intelligence (AI) and machine learning (ML), are introduced. The chapter then describes how edge computing can extend machine learning to the periphery of the IoT, and why this is necessary. The chapter takes a closer look at on-device machine learning, a category of Edge AI, and at voice-activated Edge AI, an application domain at the forefront of adopting Edge AI. Together, the sections of this chapter provide the necessary technical background knowledge to locate this research in a fast-developing technology landscape.

This chapter draws on the following publications:

1. W. Hutiri and A. Yi Ding. Towards Trustworthy Edge Intelligence: Insights from Voice-Activated Services. In *2022 IEEE International Conference on Services Computing (SCC)*, pages 239–248. IEEE Computer Society, 2022. ISBN 978-1-6654-8146-5. doi: 10.1109/SCC55611.2022.00043 [129]
2. W. Toussaint and A. Y. Ding. Machine Learning Systems in the IoT: Trustworthiness Trade-offs for Edge Intelligence. *2020 IEEE 2nd International Conference on Cognitive Machine Intelligence (CogMI 2020)*, pages 177–184, 2020. doi: 10.1109/CogMI50398.2020.00030 [286]
3. W. Hutiri, A. Y. Ding, F. Kawsar, and A. Mathur. Tiny, Always-on and Fragile: Bias Propagation through Design Choices in On-device Machine Learning Workflows. *ACM Transactions on Software Engineering and Methodology*, 4 2023. ISSN 1049-331X. doi: 10.1145/3591867 [131]

2.1. Introduction

The pursuit of smart systems and the privacy challenges that cloud-based processing of personal and sensitive sensor data present motivate for the adoption of Edge AI. This chapter contextualises the thesis by providing foundational knowledge on the technical concepts and technological components that constitute Edge AI, and on the particular category of Edge AI systems that are the focus of this work. These systems motivate the need for developing design patterns to detect and mitigate bias, and the choice of use cases in which this thesis validates the patterns.

The chapter starts by introducing the Internet of Things (IoT) and edge computing, two technological paradigms that enable sensor data collection and processing, in Section 2.2. It then provides a brief overview of Artificial Intelligence (AI) and advanced data processing techniques with Machine Learning (ML) in Section 2.3. These sections cover the foundational technologies of Edge AI. Edge AI shifts ML computations from centralised cloud servers to the edge of the IoT. Edge AI and on-device ML, a class of Edge AI systems that executes ML computations directly on devices, are further elaborated on in Section 2.4 From there, the chapter steps into the realm of applications. Section 2.5 presents an overview of voice-activated Edge AI, which underpins the use cases studied in Chapters 5 and 6. Finally, Section 2.6 reflects on the chapter and leads into the related works of Chapter 3.

2.2. The Edge of the Internet of Things

The Internet of Things

Developing and deploying smart systems is a complex undertaking that requires the coordination of multiple layers of interacting stakeholders and technologies to enable environmental perception, communication and intelligence [168]. A key technological enabler of smart systems is the Internet of Things (IoT). The ambition behind the IoT is to extend the digital realm of the Internet and the Web to the physical world of objects [102]. Definitions of the IoT vary, but agree on the perspective that the IoT consists of uniquely identifiable physical objects (or "things") with a virtual representation [320]. Objects can have the capability of knowing their precise location, obtaining data about their state or the environment and of modifying the environment through remotely controlled actuation [132]. Additionally, objects can exchange and process data according to agreed schemes and standards.

Data processing, while optional, is necessary to transform a network of sensors that perceive the environment into a smart system that can inform human decision makers or control the environment. As data processing capabilities of objects are often limited, IoT sensor data is typically transferred to remote cloud servers to be processed and analysed. However, the cloud-based processing paradigm has several drawbacks: data providers lack control and ownership over their data, privacy and security of sensitive

data cannot be guaranteed, and data transfer over wireless communication channels introduces latency and bandwidth constraints [332].

Edge Computing

In the IoT, decentralised and geographically distributed computing resources that are located at the periphery of the Internet are called the edge [63]. The edge enables data processing closer to the points of data collection (see Figure 2.1), which reduces or even eliminates the need to send data to centralised cloud servers [303]. The edge presents an alternative paradigm for processing data in IoT-enabled smart systems. It can offer better privacy, lower latencies and offline operation through localised computations. Edge computing can be defined as the enabling technologies that allow computation to be performed at the edge, including any computing and networking resources along the path between data sources and cloud-based data centres [266]. When it comes to user privacy and the protection of personal information, localised data processing with edge computing overcomes an important limitation of cloud-only systems.

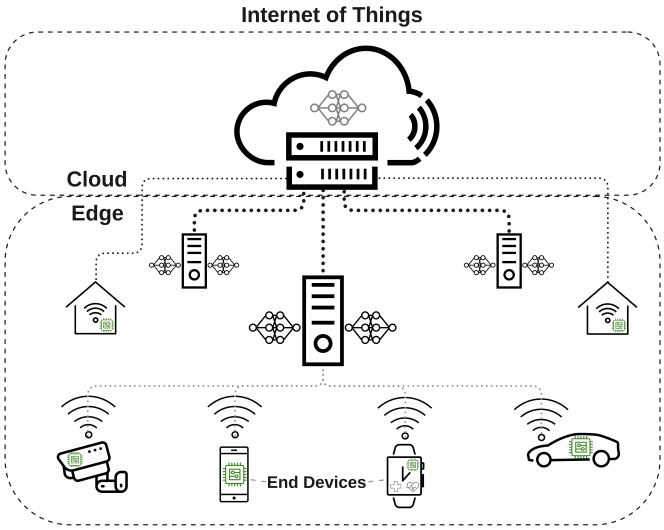


Figure 2.1: The Internet of Things (IoT), separated into Cloud and Edge computing and with end devices ("things") that collect and sometimes process sensor data.

Edge computing technologies have varying processing capabilities and connectivity [254]. Closest to the data source are end devices which perceive and control the environment with sensors and actuators. End devices often have embedded microcontrollers that can be exploited for computation. However, processing and memory resources of end devices are scarce and power consumption is severely restricted, especially if devices are battery operated [82]. Gateways appear next in the networking

hierarchy to settle the heterogeneity between diverse protocols of different networks and the Internet. They typically have more computational power than end devices and can be used to perform some data processing. A further level up are fogs, which are servers with advanced data processing capabilities, large memory resources and stable electrical power connections. Fogs extend the capabilities of the cloud closer to end users by distributing computation, communication, control and storage [39].

2.3. Artificial Intelligence and Machine Learning

Artificial Intelligence

Artificial intelligence (AI) emerged as a research discipline in the middle of the 20th century [248]. Over the past decade AI has moved from lab experiments into many real world products and systems [69]. It has received widespread public attention and encompasses so many fields of specialisation that its meaning is often implicitly assumed. However, AI can be considered from different perspectives, and it is useful to position this thesis in relation to them.

Russel and Norvig [248] point out four aspirations for research in AI: to build systems that think like humans, that act like humans, that think rationally or that act rationally. To build systems that think like humans, human cognitive processes need to be understood and mimicked. Systems that think rationally, on the other hand, are built to reason logically. In contrast to "thinking" systems, building systems that act implies that AI systems should display behaviour that either mimics that of humans or that follows logical reasoning. This thesis studies systems that collect observational data from the environment with sensors in the IoT in order to mimic human sensory perception, and thus aligns most closely with the paradigm of building AI systems to "act like humans".

Framing AI systems as *built* systems has been particularly influential for this thesis. Simon [268], an early pioneer in AI research, emphasised the implications of human built artifacts by distinguishing the natural world (and natural sciences) from the artificial. Objects and phenomena in the natural world exist independently of human interventions. The artificial, however, is a world of artifacts and systems invented and created by humans to achieve their goals and purposes. Positioning AI as systems built by humans and acknowledging their artificial nature entails that they are designed. This then makes careful contemplation and study of how they are designed a worthwhile research endeavour. This thesis views AI systems as technological artifacts that are purposefully created. It also takes the normative stance that AI systems should satisfy collective human goals and purposes that benefit society, the environment and other intelligences sharing the planet with humans.

Viewing AI systems as technical artifacts only offers a partial perspective on how they are designed. An expanded view of AI systems considers them as complex socio-technical systems [301]. The study of complex socio-technical systems extends system

boundaries beyond those of technical artifacts to include their interaction with human agents, and with aspects of organisations or institutions that determine explicit or implicit rules of engagement between human agents and technical artifacts [17, 53]. Considered from the socio-technical systems perspective, this thesis studies the interaction of technical artifacts that constitute AI systems with human agents that desire to build them to benefit society. Nonetheless, for clarity of communication I will refer to the technical artifacts as AI or AI systems, and constrain human agents to developers or engineers that build AI systems.

Machine Learning

Machine learning (ML) describes a broad class of AI systems that use statistical learning algorithms to fit functions over data to discover patterns and correlations [110]. Typical tasks that ML systems perform are clustering, classification and regression. These tasks are used to group data based on characteristic attributes, to detect patterns and anomalies in the data, to make recommendations, discover trends, make forecasts, and emulate audio and visual perception. ML systems learn models by iteratively approximating functions that transform input variables in a given dataset to an output value. Once such a function has been approximated, the model can be used to predict an output value for new input variables provided that the input and output come from the same distributions as the dataset used to approximate the function. The process of learning to fit a function over data is called model training, while using a model to make predictions is called inference [8].

The two main categories of algorithms used in ML are supervised and unsupervised learning algorithms [110]. Supervised learning requires that each data input during training is labelled, meaning that it has a known output value. During supervised model training the labels (i.e. the output values) are used as a guide to find a parameterised function that minimises the error between the model's predicted output values and the known output labels. In supervised learning, the performance of a model is evaluated based on its ability to predict a correct output for a new data input. On the contrary, the goal of unsupervised machine learning is to discover structure in the input data if no labels are available and the output is unknown.

ML methods scale to very large datasets and improve with more data [198]. They have thus become essential for processing the extreme quantities of data produced by digital, online services and applications [138]. At the same time, ML systems have also benefited and been enabled by the continuous data streams produced by digital and online services. A particular type of supervised learning that has gained from large, online datasets is deep learning. Over the past decade these multilayered, neural network based techniques have continued to out-compete other approaches and provide state of the art predictive performance for many perception tasks [165]. Models trained with deep learning algorithms are called deep neural networks (DNNs), and the structural components of a DNN model are called its architecture. Popular DNN architec-

tures are convolutional neural networks (CNNs) [105] and recurring neural networks (RNNs) [223]. In the IoT, DNNs have been very successful at processing the massive volumes of data generated by sensor systems [201, 310, 311]. They have been rapidly adopted and are now widely used to process text, image, video, speech and audio data.

2

Machine Learning Workflows

ML development can be visualised as a data processing workflow as shown in Figure 2.2. To build a ML model, training data needs to be gathered and stored. Once relevant and sufficient data is available, it is then pre-processed to be transformed into an input that is appropriate for model training. A model can be deployed after training if an evaluation deems its performance as sufficient. During deployment the model is used for inference to predict output values for new data inputs. For many models, model parameters can be optimised alongside model training to better suit the data [110].

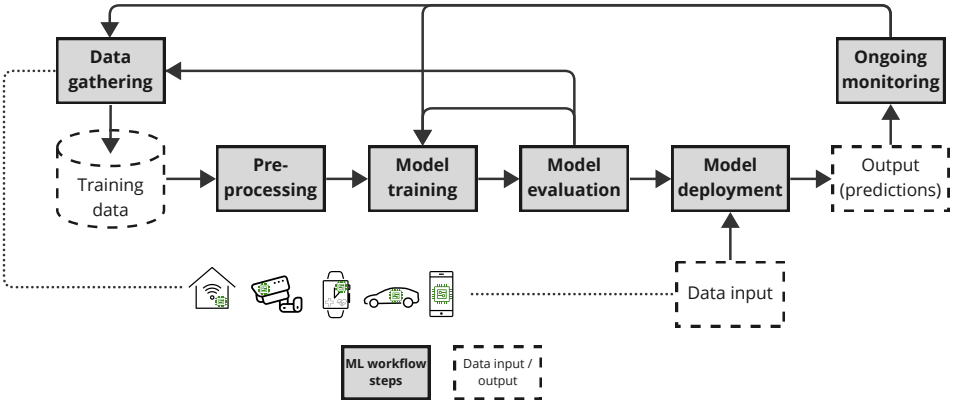


Figure 2.2: A typical ML development workflow

Training a model with high performance (i.e. low error) depends on many aspects of the ML workflow, in particular the training data, the optimisation algorithm and the evaluation functions [65]. However, model performance is not constant over time. As the deployment environment evolves and changes, so does the data input and the model performance may deteriorate as a result. Models thus need to be monitored and updated constantly [256]. This can be done by retraining a model with new data, or by using online learning algorithms over streaming data [92]. The dependencies and feedback loops between data processing steps make managing ML workflows a complex undertaking [259]. In addition to changes in the deployment environment, unexpected challenges can arise when the long-term maintainability of ML systems is not considered during development.

2.4. Edge AI: Extending ML to the Edge of the IoT

Edge AI broadly encompasses the distribution and execution of data processing workloads on and for the edge [55]. Key benefits of using Edge AI to process IoT data are reduced latency, lower bandwidth requirements and improved data privacy [228, 332]. These benefits are particularly relevant when network connections are intermittent or systems need to work offline, and when IoT data contains personal or sensitive information. While data processing is the primary goal of Edge AI, it also requires hardware, software and networking components. Considering the success of ML and DNNs in processing sensory IoT data, Edge AI is largely concerned with the training of or inference on ML models. This section presents an overview of the topologies that distribute ML workloads across computing resources, followed by an introduction to on-device ML, a specific Edge AI topology that is studied in this thesis.

Edge AI Topologies

Model training and inference in Edge AI can be distributed across different computing resources like end devices, gateways, fogs or the cloud [254]. Research and engineering challenges in Edge AI vary based on where data processing workloads are executed. Device-centric computation has to overcome the limitations of hardware resources like memory, compute and battery power constraints. Consequently, it also needs to consider whether and when to offload computation to more powerful computing resources. Gateway-centric computation requires wireless communication, which can introduce unpredictable latencies that affect availability and network service quality. Fogs provide greater computational power than devices and gateways, and less latency than transmitting data to the cloud. Finally, the cloud offers unlimited storage and processing resources, but comes with high data transfer demands and communication overheads.

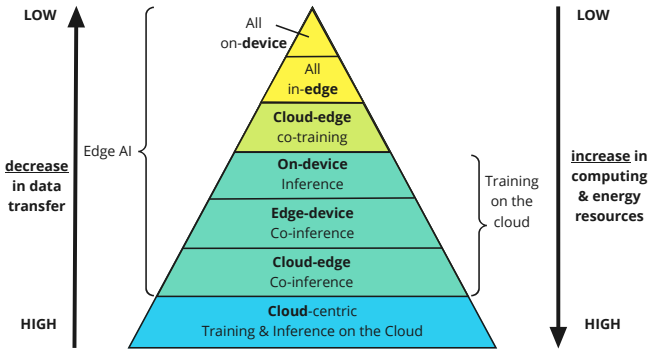


Figure 2.3: In Edge AI, ML training and inference workloads are distributed across the cloud, edge and devices. Adapted from [336]

Zhou et al. [336] present a useful overview of Edge AI topologies, which is replicated in Figure 2.3. From the Edge AI viewpoint, cloud-only computation is centralised on servers with near limitless resources, whereas Edge AI is decentralised and distributes training and inference workloads across heterogeneous fogs, gateways and devices. For ease of reference gateways and fogs are jointly referred to as edge servers. Moving from the bottom to the top in Figure 2.3, computing resources and data transfer decrease as inference and training move from the cloud to the edge and then devices. In co-inference and co-training scenarios, data processing is done cooperatively between cloud and edge servers or between edge servers and devices. An example of co-training is federated learning, which trains models across devices, edge servers and the cloud [142]. Another common topology is on-device inference, which performs resource-intensive training on the cloud, and then downloads trained models to devices for local inference [280].

On-device Machine Learning

On-device ML encompasses techniques that train models and do inference directly on end devices [38]. This can entail learning ML models on the end device, partially learning models on the device but offloading some computations to more powerful edge servers or the cloud, retraining pre-trained models on-device, or doing inference on-device. The most common approach to on-device ML, and the approach considered in this thesis, is to combine cloud training with on-device inference [60]. This approach leverages the abundant resources of the cloud for the computationally intensive training tasks, while minimising latency and ensuring data privacy during on-device inference.

On-device inference is driven by constraints, as it needs to take the limited memory, compute and energy resources of end devices into account [14]. The available storage and runtime memory on a device limit the size of the ML models that can be deployed on it. The execution speed of inferences on the device is directly tied to the available compute resources. Moreover, the amount of computations required by a model has a direct relation to its energy consumption. Given that many end devices are battery powered and have limited energy resources, it is essential that on-device ML models operate within a reasonable energy budget. In addition to these resource constraints, on-device inference also has to deal with variations in the hardware and software stacks of heterogeneous end devices. For instance, prior research [186] has shown that different sensor-enabled devices can produce data at different sampling rates owing to their underlying sensor technology and real-time system state. Such variations can impact the quality of sensor data input to the ML model, which in turn can impact its prediction performance.

Research in on-device ML is largely concerned with overcoming these constraints and satisfying hardware-based performance metrics while achieving acceptable predictive performance [60]. Prior works have developed interventions to overcome mem-

ory and compute limitations, like weight quantization [107] and pruning [174]. Other approaches such as input filtering and early exit [126], partial execution and model partitioning [59] allow for dynamic and conditional computation of the ML model depending on the available system resources. Another common approach to satisfying resource constraints is to design light-weight architectures that reduce the model footprint [31, 329]. Finally, solutions have been proposed to make ML models robust to different resolutions of the input data [203], which is a key to dealing with sampling rate variations in end devices. Common to all these interventions is that they trade-off a model's resource efficiency with its prediction performance. For example, model pruning or the use of light-weight neural architectures can result in a model with smaller memory footprint and faster inference speed, however it comes at the expense of a slight accuracy degradation [31, 174, 329].

2.5. Voice-activated Edge AI

From voice assistants and conversational agents, to social robots and avatars, the voice is an important interface for humans to communicate and interact with digital services [261]. Voice assistants such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana enable verbal, hands-free and eye-free interaction with web services (e.g. asking about the weather), personal information (e.g. retrieving calendar information) and smart home devices (e.g. turning on the lights). Despite the popularity and large-scale adoption of voice-based services, data privacy and security remain an ongoing concern [72, 164, 258, 262]. Edge AI is thus becoming increasingly important to process voice data in voice assistants [21, 47].

The seeming simplicity of voice interactions is made possible by a complex system of hardware, software, networked communications, machine learning and voice assistant skills. Together with their human and institutional stakeholders, these components constitute the voice-based services ecosystem. Figure 2.4 illustrates the technical components that activate voice assistants and process voice data to provide services in a stereotypical voice-based ecosystem. Data processing and storage tasks are distributed across three layers: at the device level, voice assistants are activated with wake-word detection or keyword spotting on a smart device. Once activated, the device transmits the recorded voice signal to a cloud service provider. Here the voice signal undergoes advanced processing to authenticate and distill the intent of the user. The intent is used to formulate a query, which often invokes a third-party service to retrieve the requested information. The query response is sent back to the cloud service provider, which synthesises a spoken response that is transmitted to the device and returned to the user.

Voice activation constitutes the technical components responsible for provisioning and securing access to voice-enabled services. This includes activation components, namely wake-word detection and keyword spotting, and authentication components,

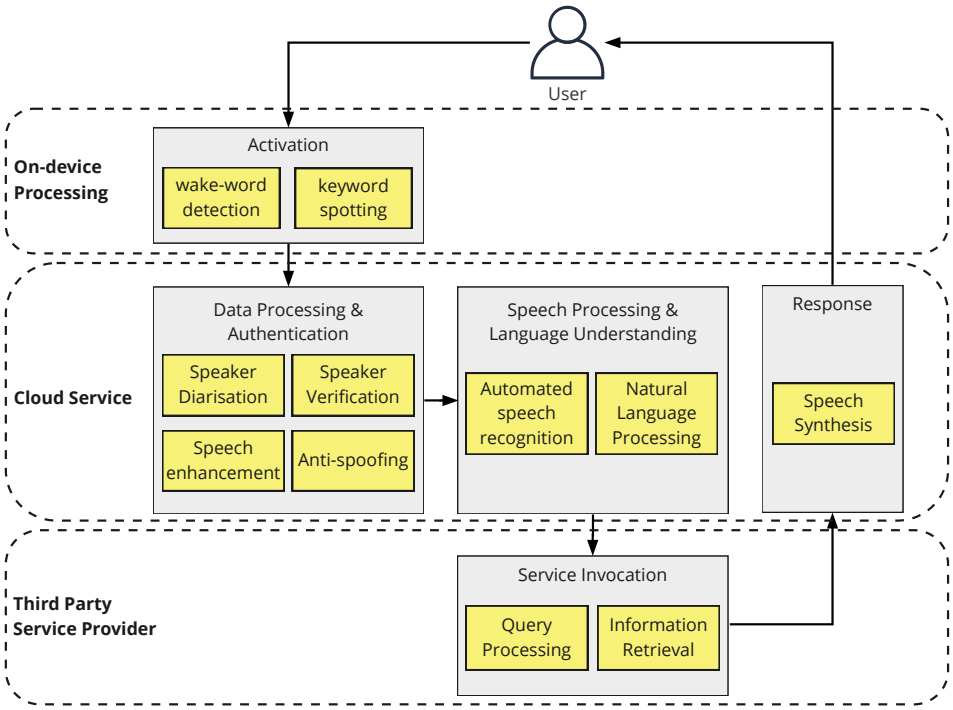


Figure 2.4: Voice activation and processing in voice assistants

which include speaker diarisation, speech enhancement, speaker verification and anti-spoofing. Wake-word detection and keyword spotting typically use on-device ML to reduce response latency and enhance user privacy. Speaker verification is a voice-based biometric that serves an important security function in the system. Anti-spoofing aims to prevent adversarial attacks on speaker verification. Speaker diarisation and speech enhancement are necessary for authentication. Together, these components are important as they directly impact whether a user has access to voice-activated services, and if this access is secure and private.

Given the pervasive deployment of voice-based services and the personal nature of voice data, voice assistants present a relevant application context to investigate aspects of Edge AI. The use cases in this thesis thus focus on the data processing components of the voice activation subsystem in voice assistants, namely speaker verification (Chapter 5) and on-device keyword spotting (Chapter 6).

2.6. Conclusion

This chapter has reviewed literature that provides technical background information for the development of design patterns in this thesis. The chapter started with an

overview of edge computing in the Internet of Things (IoT) and continued with a summary of artificial intelligence (AI), machine learning (ML) and ML workflows. It then introduced Edge AI, which distributes ML data processing from the cloud to edge and end devices. The edge enables data processing closer to the source of data collection, which reduces or even eliminates the need to send data to centralised cloud servers. When it comes to user privacy and the protection of personal information, Edge AI thus offers a promising alternative to cloud-only data processing.

The chapter highlights on-device ML as a particular Edge AI topology that is commonly used to process IoT sensor data privately, with low latencies and in an offline manner. However, it also emphasised that the memory, compute and energy constraints of on-device ML result in design trade-offs to balance efficiency and predictive performance. This thesis postulates that these trade-offs not only affect efficiency and predictive performance, but that they may also lead to performance disparities between groups of users. Such performance disparities can result in technology bias, and unfair or discriminatory systems.

The chapter concluded with an overview of voice-activated Edge AI to contextualise the use cases studied in this thesis. Voice-activated Edge AI is widely deployed in voice assistants and conversational agents. It makes extensive use of on-device ML for keyword spotting to invoke downstream services and also uses ML components to secure the system with speaker verification from intruders. Bias in voice-activated systems is a concern, as they are frequently used in domains like home, health and elderly care where they invoke critical services such as emergency response. It is thus necessary to detect and mitigate bias in the development of voice-activated systems. The use cases in this thesis investigate bias in speaker verification and on-device keyword spotting. The next chapter will review literature on trustworthy AI, bias and design patterns to ground the thesis in related work.

3

Related Work

This chapter moves beyond the technical foundations of Edge AI to the societal requirement that Edge AI also ought to be trustworthy. The chapter grounds the thesis in justificatory theory of trust and trustworthiness, bias, approaches to detecting and mitigating bias, and design patterns. In reviewing related work, the chapter highlights several gaps in current research. Firstly, current perspectives on trustworthiness differ between the AI and IoT communities. Specifically, diversity, non-discrimination and fairness, which are important aspects of trustworthy AI, are not considered in the IoT. Likewise, they have not been studied in Edge AI applications. As bias is a sources of unfairness and discrimination in AI systems, the chapter proceeds to review current approaches for detecting and mitigating bias from the ML fairness literature, highlighting design and implementation challenges of current approaches. Finally, the chapter reviews design patterns, which are commonly used in software engineering, as a promising tool for overcoming implementation and knowledge transfer challenges when adopting approaches from ML to detect and mitigate bias in new domains.

This chapter draws on the following publications:

1. W. Hutiri and A. Yi Ding. Towards Trustworthy Edge Intelligence: Insights from Voice-Activated Services. In *2022 IEEE International Conference on Services Computing (SCC)*, pages 239–248. IEEE Computer Society, 2022. ISBN 978-1-6654-8146-5. doi: 10.1109/SCC55611.2022.00043 [129]
2. W. Hutiri and A. Y. Ding. Bias in Automated Speaker Recognition. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, pages 230–247, Seoul, Republic of Korea, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533089 [128]

3.1. Introduction

The primary goal of this chapter is to explicate justificatory knowledge that underlies the development of the design patterns developed in this thesis. Furthermore, the chapter positions the relevance of the research in relation to the aspiration of developing trustworthy Edge AI. This is done by grounding the research in existing theoretical work on trust and trustworthiness, bias, approaches to detecting and mitigating bias, and design patterns.

The chapter starts with exploring trust and trustworthiness concepts in AI and the IoT and highlights gaps between perspectives in the two fields in Section 3.2. In particular, diversity, non-discrimination and fairness are identified as aspects of trustworthy AI that are inadequately accounted for in IoT trustworthiness. Section 3.3 examines bias and its relation to fairness and discrimination, thus motivating why detecting and mitigating bias is important for developing trustworthy Edge AI. Section 3.4 extends the review to literature on detecting and mitigating bias in ML. The section takes algorithmic and socio-technical interventions into account, and highlights implementation challenges that practitioners experience when trying to detect and mitigate bias in practice. Finally, Section 3.5 describes design patterns and their use in software engineering and AI. The chapter concludes in Section 3.6.

3.2. Trustworthy Edge AI

Trust and trustworthiness have been studied and formalised in many domains, including AI [88], the IoT [327], cyber-physical systems (CPS) [48], and digital services [190]. This section develops a theoretical basis for trustworthy Edge AI, based on the premise that trustworthy Edge AI should satisfy the trustworthiness requirements of its constituent technology components, AI and the IoT.

Trust and Trustworthiness

Trust is "a measure of confidence that an entity or entities will behave in an expected manner" [264, p. 2]. While the interpretation of trust varies across disciplines, it is generally accepted to be relational, seldom unconditional, and a judgement that is expected to inspire a course of action [167]. Trust presupposes that someone is willing to take an action, contingent on another party, under the belief that they will not be betrayed due to ill intentions or lack of competence of the other party. Trust can thus inspire behaviours such as risk taking [187] or actions that promote job performance [44]. Trust is warranted if the other party is deemed to possess attributes that make them worthy of being trusted and trust judgments reflect beliefs about the trustworthiness of the other party.

While trust is a judgement based on expectation, trustworthiness is considered a

necessary, but not the only condition for choosing to trust someone [44][1]. Trustworthiness focuses on the attributes of a trustee (the party being trusted) that justify trust. Factors of trustworthiness that have been established in the management literature consider the ability, benevolence and integrity of a human trustee [187]. In political science, Levi and Stoker [167] consider trustworthiness attributes along two dimensions: intention and competence. In their words, "the trustworthy will not betray the trust [bestowed upon them] as a consequence of either bad faith or ineptitude" [167, p. 476]. A trustee can be trustworthy, meaning that they possess the attributes that give a truster (the party that is trusting) confidence that their trust will not be betrayed, irrespective of whether trust is required or not.

Trust-in-technology research extends trust beyond social systems to non-human, artificial entities [162]. Technologies vary in their perceived "humanness", and research has found that people trust technologies differently based on this. If the perceived humanness of a technology is high, then human-like attributes such as ability, benevolence and integrity are good indicators of trust. Congruently, if the perceived humanness of a technology is low, then system-like attributes such as robustness, reliability and functionality are more appropriate indicators. These various trustworthiness attributes are reflected in conceptualisations of trustworthy AI and IoT, however, the two disciplines concretise them in different ways.

3.2.1. Trustworthy AI

The rapid advancement of AI, accompanied by harmful AI failures [200, 321], prompted the assembly of trustworthy AI expert groups [80], special interest groups [79], the development of public and private sector AI ethics guidelines [88], and large scale research collaborations to advance the state of trustworthy AI [278]. Consequently, many perspectives on trustworthy AI have been offered [34, 158, 285]. On a high level, this thesis adopts the perspective on trustworthy AI of the European Union (EU), as laid out in the EU AI Ethics Guidelines [119]. The EU AI Ethics Guideline are driven by ethical and robustness requirements and offer general guidance for building trustworthy AI. While the guidelines aim to provide guidance for operationalising ethical principles for trustworthy AI, they are aspirational in nature, and leave the development of concrete design considerations and specifications open to interpretation. Table 3.1 summarises the trustworthy AI attributes described in the EU AI Ethics Guidelines.

As this list of guidelines shows, trustworthy AI comprises attributes beyond the technical realm, for example data governance needs to consider the interaction between (multiple) human agents, (multiple) organisations and technical capabilities and requirements [134]. Furthermore, the predictive and decision-making capabilities of learning-based AI systems like machine learning (ML) are contingent on data from which the system can learn, and a data-processing pipeline that specifies and performs

[1] In addition to trustworthiness, the trust-propensity of the truster is also important.

AI attributes	Descriptions
Human agency & oversight	Supporting human autonomy and decision making, and promoting a flourishing, democratic and equitable society
Technical robustness & safety	Ensuring physical and mental integrity of humans, and reliable system behaviour that minimises and prevents unintentional, unexpected and unacceptable harm, even under uncertain or adversarial operating conditions
Privacy & data governance	Protecting the fundamental right to data privacy, including aspects of data quality, integrity, relevance, access and processing
Transparency	Communicating system capabilities, purposes and business models openly, making data processing traceable, and decisions explainable so that they can be contested
Diversity, non-discrimination & fairness	Ensuring inclusion and diversity throughout the AI system life cycle, inviting stakeholder participation, and designing for accessibility to ensure equal access and avoid unfair bias
Societal & environmental well-being	Promoting benefit for all human and sentient beings, future generations, society at large, and the environment
Accountability	Subjected to scrutiny and redress through auditing and reporting, and consideration of trade-offs posed by trustworthiness concerns

Table 3.1: Attributes of Trustworthy AI based on the EU AI Ethics Guidelines[119]

the learning (i.e. model training). This means that AI trustworthiness must be evaluated on a continuous basis and throughout the ML workflow (see Section 2.3) [27, 277, 285]. While trustworthy AI attributes span across a broad spectrum of stakeholder values, it may not be possible to develop systems that meet all values simultaneously.

3.2.2. Trustworthy IoT

Within the Edge AI paradigm, the IoT and Cyber Physical Systems (CPS) can be considered from a unified perspective[2] and this thesis jointly refers to them as IoT. Trustworthy IoT includes attributes of privacy, reliability, resilience, safety and security as described in Table 3.2. These trustworthiness attributes serve to assure that systems behave as expected under various operating conditions. The attributes interact and are interdependent. They affect not only each other but also other IoT concerns. Interdependencies between attributes raise challenges for trustworthiness, for example the interaction between software and hardware can result in programming bugs that drain the batteries of a critical component, or components developed by different institutions need to be and remain compatible over time [247].

[2]This view is motivated by the steady convergence of the two fields, and the benefits of advancing research progresses in both domains through a unified perspective [102]. Moreover, IoT and CPS communities view trustworthiness similarly (see for example the US NIST CPS Framework [48] and challenges and opportunities for trustworthy AI published by the Industrial IoT Consortium [28]).

IoT attributes	Descriptions
Privacy	Preventing entities from gaining access to data stored in, created by, or transiting the IoT, in order to mitigate risks associated with the processing of personal information
Reliability	Delivering stable and predictable performance in expected conditions
Resilience	Withstanding instability, unexpected conditions, and gracefully returning to predictable, but possibly degraded, performance
Safety	Ensuring the absence of catastrophic consequences on the life, health, property, or data of stakeholders and the physical environment
Security	Ensuring that all processes, mechanisms and services are internally or externally protected from unintended and unauthorized access, change, damage, destruction, or use. Considers confidentiality, integrity and availability.

Table 3.2: IoT trustworthiness attributes and definitions from the NIST CPS Framework [48]

3.2.3. Aligning Trustworthy AI and IoT Perspectives

Using the definitions of trustworthiness in Tables 3.1 and 3.2 as a theoretical foundation, Table 3.3 compares trustworthy AI and IoT attributes. The attribute of *robustness and safety* in trustworthy AI aligns most readily, and spans across several trustworthy IoT attributes. The need for "reliable system behaviour" speaks to IoT *reliability*, performance under "uncertain" operating conditions relates to IoT *resilience*, "minimising and preventing harm" translates to IoT *safety* concerns and "adversarial operating conditions" affect IoT *security*. In both domains, the *privacy* attribute is focused on protecting the right to data privacy and the processing of personal information. In addition, *privacy* in trustworthy AI also includes *data governance* and aspects of data quality, integrity, relevance and access. *Privacy* in trustworthy AI thus also aligns with the *security* attribute in trustworthy IoT.

At first glance, trustworthy AI attributes other than *robustness and safety* and *privacy* do not overlap with those of IoT trustworthiness. However, on closer examination the aspirations of trustworthy AI attributes can be mapped to IoT concerns that relate to other (i.e. non-trustworthiness) aspects. For example, the *diversity, non-discrimination and fairness* attribute of trustworthy AI will influence *human factors* and *usability*, which are part of the *human* aspect in IoT. They also relate to *constructivity*, which is concerned with how the *composition* of modular components satisfies user requirements. Similarly, a lack of *transparency* and *accountability* mechanisms on the side of AI systems will make it difficult for authorised entities to gain and maintain awareness of the state of Edge AI services, thus reducing their *monitorability*.

Apart from considering alignment between concepts, the same concepts can mean different things in the two domains. For example, a fairness-aware framework for crowdsourcing IoT energy services considers fairness as an optimisation problem, with

Trustworthy AI Attributes	Trustworthy IoT Attributes					Alignment with definitions of other IoT concerns
	Privacy	Reliability	Resilience	Safety	Security	
Agency & oversight						Manageability, Monitorability, Discoverability, Operability
Robustness & safety		X	X	X	X	States, Uncertainty
Privacy & data governance	X				X	-
Transparency						Communication, Monitorability, Enterprise, Quality, Utility, Operations on data, Relationship between data, Responsibility, Complexity, Discoverability
Diversity, non-discr. & fairness						Constructivity, Human factors, Usability
Well-being						Environment
Accountability						Measurability, Monitorability, Regulatory, Responsibility, Discoverability

Table 3.3: Alignment of definitions of Trustworthy AI and IoT attributes (X = alignment).

the goal of maximising the use of energy services across a time period [160]. This perspective diverges from *fairness* in AI, which is concerned with inclusion, diversity and discrimination.

3.2.4. Towards Trustworthy Edge AI

Neither trustworthy AI, nor trustworthy IoT attributes address the full spectrum of trustworthiness concerns that arise in Edge AI. Moreover, to build trustworthy Edge AI, interactions, interdependencies and trade-offs between AI and IoT components must be considered, as failures of AI trustworthiness may affect a variety of IoT aspects. Consider, for example, an on-device keyword spotting (KWS) system as shown in Figure 2.4. A user may attempt repeatedly to activate the system if its predictive performance is poor. This increases the computational load, which leads to increased power consumption and faster drainage of the device battery. If the ML component of the KWS system also lacks on the *diversity, non-discrimination and fairnesss* attribute, the quality of predictive performance is not random, but can be attributed to a user's personal attributes, like their age. Users could then experience disparate hardware performance (e.g. the longevity of their device battery), or service quality (e.g. their ability to access emergency response), as a consequence of the ML component favouring people with a particular personal attribute over others.

As this example demonstrates, it is important to consider trustworthy Edge AI comprehensively, taking trustworthy AI and IoT considerations into account. Yet many recent roadmaps and reviews of Edge AI focus only on trustworthy IoT attributes [55, 98,

193], and trustworthy Edge AI has been studied primarily from the perspective of IoT trustworthiness [214, 275, 294]. This thesis thus posits the trustworthy AI attribute of *diversity, non-discrimination and fairness* as a prerequisite for trustworthy Edge AI. While some literature has pointed out that bias and fairness in IoT data analytics are an open problem [26, 64], it is poorly understood how this trustworthy AI attribute is affected by the resource constrained and highly localised settings of Edge AI applications. This motivates the study of inclusive, non-discriminatory and fair Edge AI as a relevant academic endeavour.

3.3. (Un)Fairness, Discrimination and Bias in ML

Having laid out the need for trustworthy Edge AI in the previous section, and in particular for Edge AI that enables *diversity, non-discrimination and fairness*, this section clarifies the key concepts of fairness, discrimination and bias, and discusses bias as a source of unfairness and discrimination in machine learning (ML).

3.3.1. Concept Clarifications

Bias, discrimination and unfairness are frequently used interchangeably to describe decision making processes and systems that favour or are prejudiced towards an individual or a group of people based on their intrinsic or acquired attributes [192]. Despite their proximity in meaning, it is important to distinguish between the concepts, as they do not necessarily entail one another.

Fairness

Fairness is a cross-cultural phenomenon in human societies [23]. People care about being treated fairly and treating others fairly. Even children have a sense of what it means to be fair [189]. One theory for the development of human fairness is that it evolved to support cooperation [23]. This may be particularly true when considering fairness in relation to resource allocation or the application of distributive justice [189]. Here, notions of fairness can be regarded from two lenses. Firstly, individuals enforce fairness to resist being disadvantaged themselves. However, individuals have also been shown to avoid accumulating personal advantage if that implies putting others in a position of disadvantage.

Empirical studies show that people are sometimes willing to resist unfair actors even if it comes at a cost to themselves [141, 299]. In studies of organisations, fairness has thus been used to explain opposition to excessive monopoly gains and price discrimination. Furthermore, people apply systematic but implicit rules that specify which actions are considered unfair. However, these fairness rules do not necessarily follow expected theories and intuition, and should be established with empirical data. The notion of fairness is not limited to the acquisition and ownership of material

goods. It can also be considered from the perspective of equality [100], and has been argued to be a moral virtue in its own right [299].

Discrimination

The Merriam Webster dictionary defines discrimination as a "prejudiced outlook, action or treatment" or the "act, practice or an instance of discriminating categorically rather than individually" [194]. In these two definitions it is worth noting the emphasis on action, treatment and practice, implicating that discrimination has a propensity to result in prejudiced behaviour [66]. Article 2 of the Universal Declaration of Human Rights declares that it is a universal human right not be subjected to discriminatory behaviour:

"Everyone is entitled to all the rights and freedoms, without distinction of any kind, such as race, color, sex, language, religion, political or other opinions, national or social origin, property, birth, or another status. Furthermore, no distinction shall be made on the basis of the political, jurisdictional, or international status of the country or territory to which a person belongs."

In many countries anti-discrimination, or non-discrimination, is a legal requirement that can be enforced by law [308]. Non-discrimination law specifies protected or sensitive attributes based on which individuals and groups of people must be treated equally in decision making processes [200]. Personal demographic attributes such as gender or age are well known protected attributes. However, relying on protected attributes as a panacea for identifying discrimination has limitations. Discrimination often happens along multiple axes of prejudice [215] (e.g. gender and ethnicity). Considering protected attributes along single axes (e.g. only gender) thus cannot reveal discrimination at the intersection of more than one protected attribute [121]. Protected attributes can also correlate with other personal attributes that may not be protected [104]. These attributes are called proxy attributes, as they can be used to indirectly discriminate [45]. Proxy attributes arise from the organisation and structure of the physical world and often reflect historic injustices. A well known example of a proxy attribute are postal codes, which reflect the demographic make-up of a neighbourhood and can thus convey information about ethnicity and socio-economic status, even if the codes themselves are considered neutral [295].

Bias

It appears ironic that despite caring about fairness, societies still need explicit guidelines that prohibit discrimination. However, the study of bias in human judgement and decision making has shown that humans are far from being rational decision makers [300]. Instead, emotional and social factors influence how we make choices [140]. Humans are easily fooled by cognitive illusions and make errors of judgement. These errors arise, at least in part, because humans take mental shortcuts and rely on heuristics when making decisions.

Tversky and Kahneman [300] identified and characterised three heuristics and resultant biases that people are susceptible to when assessing probabilities and making predictions. The *availability* heuristic focuses on information at hand. It leads to biases due to the ease with which an event or situation can be recalled from memory and the ease with which instances can be searched and imagined. The *representativeness* heuristic relies on prototypes and stereotyping, and is evident in situations where people conflate probability with similarity. It leads to biases in probabilistic reasoning, as people tend to be insensitive to the prior probability of outcomes, sample size and similar effects. Lastly, the *anchoring and adjustment* heuristic gives disproportionate weight to the first bit of information that is encountered. This information becomes a mental anchor, which biases subsequent decisions when future assessments of situations are insufficiently adjusted from the anchor.

These and other innate cognitive biases in judgement and decision making [322] distinguish real humans from economic models of 'rational man'. They align with a general definition of bias as a skewed, or slanted perspective [90], for example a person may be biased by having a preference for buying white cars. However, bias is frequently understood to also imply instances of prejudice [66]. Such biases carry significant social consequences, as they lead to decisions or actions that systematically assign undesirable outcomes, or deny opportunities or goods, to individuals or groups on grounds deemed unreasonable or inappropriate [90]. This thesis is concerned with these kind of biases that lead to prejudiced outcomes.

3.3.2. Bias in ML: A Source of Unfairness and Discrimination

Algorithmic decision making and AI have been argued to be an antidote to the cognitive biases just described [52, 94, 152, 154]. However, positioning algorithms to remedy bias raises two major concerns. Firstly, bias in decision making systems is not only attributable to isolated, formal decision making process, whether human cognition or machine prediction. Cognitive biases inform how people experience and interact with the world, and subsequently how they shape and design the world around them [67]. Biases thus do not remain mental constructs but structure and organise the world. Consequently, cognitive biases find their way back into the decision making systems in which they were meant to eradicate bias [100]. Advocating for the potential of algorithms to reduce bias without considering who holds the power to propagate biases, whose preferences are maintained and who continues to be prejudiced against, neglects accounting for the socio-technical contexts in which algorithms are created, deployed and maintained [12].

This leads to the second concern. AI systems codify and institutionalise decision making processes in a manner that enables bias to spread easily and efficiently. Almost three decades ago, Friedman and Nissenbaum [90] first studied why bias in computer systems that automate decision making is particularly problematic. Computer systems, even though complex, can attain scale at a relatively low cost. They can thus

hide biases in code, while broadcasting them to a wide user base. Biased computer systems do not necessarily disclose their biases to users, let alone offer means to contest their decisions. Thus, computer systems have the ability to scale and spread bias in automated decision making at a scale that far exceeds that of individual humans. These reasons still hold true today, as AI amplifies and accelerates the rate at which biases can scale. It is for these reasons that bias in AI remains an area of significant concern.

Risks and Harms from Bias in ML

Many studies have revealed evidence of bias in ML applications, ranging from natural language processing [25] and gender classification [29] to face recognition [240], person detection [325] and automatic speech recognition [155, 281]. The algorithmic fairness literature categorises harmful consequences of biased AI systems as allocative or representational harms [16]. Representational harms perpetuate stereotypes. For example, word embeddings in natural language processing have been found to replicate historic gender stereotypes when modelling occupations [25]. Allocative harms deny opportunity or resources to a group of people based on their inherent attributes. This has had repercussions in many decision making systems, like consumer lending [324], criminal justice risk assessments [19] and hiring [237].

Biased outcomes can lead to discrimination, but a biased system does not necessarily need to be discriminatory. For example, consider an application in which a voice assistant is intentionally designed to serve a particular user group, such as pensioners. While the decision to design for a limited user group could be cast as bias, it can be justified by the application's design requirements. On the contrary, a voice assistant that does not exhibit prejudice against users based on its technical performance can still be deployed in a manner and context that are discriminatory. Consider a second example of a call center that provides women with an opportunity to work from home. The call center agents are supported in their tasks with voice technology that performs equally well for all agents, irrespective of their accent, age, physical ability etc. However, the voice technology has the capability of recognising babies crying in the background, and thus directs calls away from agents where this is the case. Even if the technical capabilities of the system are equal across agents, it is deployed in a manner that systematically denies agents with small children the opportunity to work, and may thus be regarded as being unfair.

Sources of Bias in ML

Recent work in software engineering has highlighted the need to model quality aspects of ML systems in detail [267]. Bias is a new concern affecting ML software quality that should be considered as a non-functional requirement during development [124]. In requirements engineering and quality modeling, bias considerations are allocated to data-related aspects [267, 307]. However, like hidden technical debt in ML [259],

bias can emerge at different stages in the ML workflow, such as during data collection, population sampling and measurement. Sources of bias should thus be investigated at different stages of the ML workflow, including training data generation, the predictive model and the evaluation mechanism [197].

Suresh and Guttag [277] identify seven sources of bias in ML workflows that can have harmful consequences. *Historical bias* replicates biases, like stereotypes, that are present in the world as is or was. *Representation bias* underrepresents a subset of the population in the sample, resulting in poor generalization for that subset. *Measurement bias* occurs in the process of designing features and labels to use in the prediction problem. These three sources of bias occur in the data generation stage of the ML life cycle. Further sources of bias can emerge in the model building and implementation stage. *Aggregation bias* arises when data contains underlying groups that should be treated separately, but that are instead subjected to uniform treatment. *Learning bias* concerns modeling choices and their effect on amplifying performance disparities across samples. *Evaluation bias* is attributed to a benchmark population that is not representative of the user population, and to evaluation metrics that provide an oversimplified view of model performance. Finally, *deployment bias* arises when the application context and usage environment do not match the problem space as it was conceptualised during model development.

Feedback Loops and Design Choices

These sources of bias are rarely static, isolated instances. ML systems are complex socio-technical systems that consist of multiple human and technical components that interact in non-linear and dynamic ways. Bias and other sources of unfairness are intertwined within feedback loops that emerge through these complex interactions [58, 76, 147]. For example, online recommender systems use user interaction data to determine the popularity of search or advertising results and thus infer their relevance. Recommendations that get clicked on more are thus ranked higher in subsequent search results. Over time the recommendation system reinforces its original ranking and low-ranked results continue to remain unpopular even if they are relevant [10].

Like other socio-technical systems, ML system performance is also influenced by design decisions. Holstein et al. [122] have observed that developers can feel a sense of unease at the societal impacts that their technical decisions have, as design choices can play a significant role in propagating bias in ML workflows. For example, a study on fixed-seed training of deep learning systems has shown high variance in fairness measures if experiments consist of a single run with a fixed seed [235]. Another example is in the inherently constrained on-device ML setting, where hardware limitations require that pre-trained ML models undergo multiple post-processing steps to overcome resource limitations and anticipate context shifts. Post-processing steps like domain adaptation [270] and model compression [123] introduce additional design choices that can propagate bias.

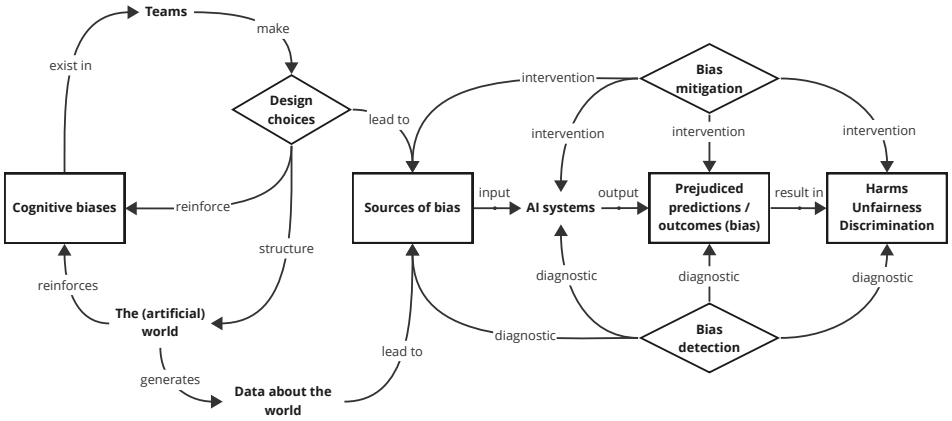


Figure 3.1: Diagrammatic representation of bias feedback loops in AI systems

In practice, designing and building AI systems that have no bias is difficult. The data streams produced by the world capture the hidden biases that are already structured in it, and cognitive biases in AI teams can lead to design choices and assumptions in the design of AI systems that replicate these biases [83]. Moreover, these effects are neither static nor linear, but create reinforcing feedback loops as shown in Figure 3.1. Despite these challenges, bias in ML systems can and should be addressed. By identifying and detecting bias, and by applying mitigating interventions, actions can be taken to manage sources of bias and prejudiced outcomes in order to prevent discrimination and unfairness.

3.4. Detecting and Mitigating Bias in ML

Many current approaches to bias detection and mitigation require that bias is quantified so that it can be measured and improved. This section provides an overview of key aspects of bias quantification that underly bias detection and mitigation approaches, with a focus on approaches used to quantify bias that can lead to allocative harms. The section then continues to review literature that highlights gaps in approaches to detecting and mitigating these kinds of biases.

3.4.1. Quantifying Bias

Allocative harms arise when prediction errors result in outcomes that systematically favour some, while disadvantaging others. From this view, bias can be framed as the unequal distribution of prediction errors that lead to biased ML system outcomes [16]. *Bias measures* or *algorithmic fairness measures* are used to quantify the extent of bias that results from disparate prediction errors or outcomes. The literature does not dis-

tinguish clearly between bias and algorithmic fairness measures and they are often used interchangeably. I consider the distinction between the terms to be one of purpose, rather than one of implementation, as I explain below.

There are two main reasons for quantifying bias, firstly to detect (or diagnose) and secondly to mitigate (or intervene). When detecting bias during model development or post-hoc when testing models and applications, *bias measures* quantify the extent of disparity due to prediction errors. Bias measures thus indicate the limits of a ML system and enable developers to interrogate the systems that they build. By contrast, when applying bias mitigating interventions, *algorithmic fairness measures* are incorporated in the optimisation procedure of an algorithm. While they also quantify bias, they impose an additional constraint on the learning process to artificially constrain the learning problem with the goal of reducing error rate disparities to make the predictor less prejudiced. Algorithmic fairness measures can be applied to mitigate bias at different stages in the ML workflow; during pre-processing, in-processing and post-processing [192].

Algorithmic Fairness Measures

Statistical fairness criteria mathematically formalise what it means for an algorithm to be fair [180]. These criteria are then used in algorithmic fairness measures to quantify disparities in prediction errors across data samples, and to enforce a reduction in performance disparities during the ML training process. Most algorithmic fairness measures are derived from disaggregated statistical base metrics, such as the true and false, positive and negative error rates [306]. Algorithmic fairness measures then calculate ratios or differences between the base metrics of a group and a reference group, or overall performance. Importantly, different algorithmic fairness measures also imply different definitions of fairness [180].

Disparities in prediction errors can be quantified to measure various things, for example if predicted positive outcomes are equal across groups of people (e.g. statistical parity), if error rates are equal (e.g. equalized odds), if similar people are treated similarly (e.g. fairness through awareness), or if the outcome of a decision making process would have been the same even if the attributes of the person were different (e.g. counterfactual fairness). These distinctions are formalised in statistical fairness criteria by the joint distribution of the protected attribute, the target variable, the predicted value and in some instances the input features [16]. While many fairness criteria have been proposed, they can be categorised as equalising one of three statistics across groups: 1) acceptance rate (i.e. a $true$ positive classification in a binary classification problem), 2) error rates, and 3) the frequency of an actual outcome given a score value. Depending which statistics they equalise, they are said to satisfy independence, separation or sufficiency respectively.

Criteria that satisfy equal acceptance rates across groups are said to require *independence*, meaning that the protected attribute must be statistically independent of

the predicted positive value. Familiar bias measures of this type are called disparate impact, group fairness, statistical parity or demographic parity. *Separation* criteria require equal error rates across groups, meaning that the predicted value must be statistically independent of the protected attribute, given the target variable. For a binary classification task this practically means that all groups must have equal true positive and false positive rates. Criteria for separation that have been proposed in the literature are equal opportunity and equalized odds [109], conditional procedure error [20], predictive equality [46] and others. Depending on the application context, it is also common to have criteria that relax some of the constraints of separation. Finally, *sufficiency* requires that the target value is independent of the protected attribute, given the predicted value. Another way of understanding sufficiency is to think of it as calibration, where a well-calibrated score (or predicted value) represents the probability that an input belongs to a particular class in the target variable, given the predicted value. A fairness criteria for sufficiency requires calibration by group. Some sufficiency criteria that have been proposed in the literature are conditional use accuracy [20], predictive parity [41] and calibration within groups [41]. Further criteria are listed in [16].

Measuring Bias in Groups versus Individuals

Bias measures can apply to groups or individuals [192]. Group-based approaches require that different groups are treated similarly. In contrast, bias quantification for individuals aims to assure that similar individuals receive similar predictions. The measures described above apply primarily to groups. Well-known approaches to evaluating bias for individuals are counterfactual fairness [159] and fairness through awareness [70]. Counterfactual fairness considers the likelihood of alternative prediction outcomes if an individual had different personal attributes. Fairness through awareness requires that that similar individuals are treated similarly. Some approaches have been proposed to combine group and individual algorithmic fairness measures, for example by extending measures with inequality indices from economics to characterise unequal algorithmic outcomes across individuals and groups [274].

There are several constraints to take into account when using group-based bias measures. Most of the measures have been developed for binary decision outcomes and binary groups (i.e. a protected and an unprotected group) [97]. In practice, this can make model evaluation and selection extremely difficult if bias must be evaluated for many models, across many groups [179]. An approach to resolving this challenge is to create *meta-measures* that aggregate bias measures across all speaker groups into a single measure to evaluate overall model bias. However, when group sizes are not equal, meta-measures can be biased themselves [179].

Even if the technical complexity of evaluating bias across groups can be resolved, group design needs to be done with care to avoid hidden stratification and bias due to categorisation and group attribute choices. Hidden stratification was first identified in medical image analysis [218] and occurs when data contains previously unknown sub-

sets of cases (i.e. groups) in which model performance is poor. Medical outcomes for patients that fall in these cases were found to be significantly worse than average. At the same time, these cases were also clinically the most severe, making hidden stratification particularly problematic. Further bias can be introduced through choices made when labelling group attributes or choosing category thresholds, for example for age or race. A study on the impact of age thresholds used to create "young" and "old" groups has shown that this choice affects not only the extent of bias measured, but also determines which group is favoured [125]. Similarly, racial category labelling has been critiqued as encoding unique racial systems that propagate stereotypes and exclude ethnicities that do not conform to them [145]. An alternative to defining groups explicitly is to consider bias by establishing comparisons to worst-case performance [97].

3.4.2. Shortfalls in Bias Detection and Mitigation Approaches

Research in algorithmic fairness has demonstrated the propensity for bias in algorithmic decision making systems. However, purely algorithmic approaches are limited in their ability to address bias. This section discusses limitations and critiques of algorithmic fairness measures, and highlights current challenges that arise when practitioners try to implement bias evaluations.

Limitations and Critiques of Algorithmic Fairness Measures

Fairness measures that satisfy *independence* are inherently limited, as true positive classifications are not the only predictions that matter in many applications. While measures that satisfy *separation* (i.e. equal error rates across groups) overcome this shortfall, they have been critiqued for use in many decision making scenarios [16, 308]. Wachter et al. [308] call these metrics bias preserving, as they rest on the assumption that the labels (i.e. target variables) of data samples used to train a decision making system accurately represent the outcomes of an unbiased decision making process. However, when label categories are distributed unevenly across groups (e.g. one group has a higher base rate for paying back loans), even an optimal predictor will not satisfy equal error rates across groups [16]. Likewise, if the labels themselves are biased, fairness criteria are ineffective at detecting or mitigating bias. In reality this is often the case, as data about the world reflects biases that exist in the world, as shown in Figure 3.1. Bias preserving metrics thus propagate historic bias, inequality and injustices, replicating a biased world-view to maintain the social status quo.

Beyond the limitations of individual measures, a conundrum arises when conflating bias with fairness, and considering fairness from a strictly algorithmic perspective. The "impossibility of fairness" [100] is a mathematical proof that algorithmic fairness measures cannot satisfy independence, separation and sufficiency criteria simultaneously [41, 153]. This conundrum cannot be resolved algorithmically, as it requires a decision to be made on what fairness objectives are appropriate for a particular application, given the consequences of the prediction outcome [45]. Yet, ap-

proaches that reduce fairness to mathematical formalisms of algorithmic fairness and "debiasing" are frequently applied without considering the application context, and the type and severity of risks and harms that can result from unfair algorithmic systems [12, 100, 121, 133]. Green [100] argues that a reform to the application of algorithmic fairness approaches is necessary to take substantive measures to address social inequality.

There are applications where measures that quantify error rates to achieve error rate parity are not bias preserving. This requires that labels can be known exactly, that no historic bias exists, that known performance disparities are legally justified or that systems are designed for the purpose of debugging [308]. In applications where there are no inherent differences between individuals or groups that would justify disparate prediction errors, measures of error rate parity are then acceptable indicators of bias. This is the case for face recognition, where error rate comparisons are a primary way of measuring bias [157, 241]. Similarly, in many on-device ML applications such as wake-word detection, keyword spotting, object detection and speaker verification, assessing bias by quantifying error rate disparities is useful.

Implementation Gaps

In their survey of open source fairness toolkits, Lee and Singh [166] identify several implementation challenges that practitioners face when trying to conduct bias evaluations. Many practitioners have a limited background in fairness, yet need to evaluate bias in specific application contexts. Moreover, practitioners typically need to translate their analysis and findings to a non-technical audience. While fairness toolkits ought to support practitioners in their efforts to detect and mitigate bias, they only offer one-size-fits-all user interfaces that are tailored towards practitioners with prior understanding of fairness. Additionally, toolkits lack consistency in their methodological approaches. This limits the accessibility of toolkits and makes learning curves steep. Moreover, most toolkits are academic prototypes that are inadequate for real-life use cases. Searching and comparing toolkits is difficult, and adapting toolkits to customised use cases is often not possible. Toolkits provide limited coverage of bias concerns in the end-to-end ML lifecycle and are not readily integrated into existing workflows. On the one hand they flood users with an information overload, while at the same time oversimplifying the analysis and fairness definitions.

Deng et al. [56] examine how ML practitioners use fairness toolkits to evaluate bias. They observe that practitioners seek convenient solutions rather than appropriate ones, as exemplified by the common practice of copying code from toy examples provided by toolkit APIs. Amongst practitioners with little or no training around ML fairness, it is common to apply a "fairness through unawareness" approach, which attempts to mitigate bias by simply ignoring protected attributes. Practitioners have expressed the desire to use toolkits as educational tools, yet toolkits only offer limited insights into details behind implemented methods. Many toolkits have unintuitive doc-

umentation, and ML fairness methods are difficult to convey to others. Moreover, as toolkit functionalities are constrained, ML problems are reformulated to a format for which current toolkits provide support. The functionalities and limitations of toolkits thus define the scope of the bias analysis. The authors recommend several points of action to improve bias evaluations. They recommend that tools provide use-specific and context-specific guidance, as well as cross-functional collaboration and organisational buy-in. Futhermore, there is a need to support learning within toolkits, to better support incorporating toolkits into existing ML pipelines, and to consider practitioners' time constraints. The authors call on toolkit developers to publish toolkit patterns and anti-patterns.

3.4.3. Supplementing Algorithmic Fairness with Design

Detecting and mitigating bias in ML applications remains challenging, despite numerous algorithmic fairness approaches that have been proposed in the literature. As highlighted in the previous section, approaches to improve ML fairness vary widely in their implementation and are frequently not matched to the actual application at hand [199]. Moreover, there is a disconnect between solutions proposed by researchers and the challenges that practitioners experience in practice [122]. These challenges motivate the need for alternative approaches to detecting and mitigating bias in ML that complement existing practices and supplement the limitations of algorithmic fairness interventions.

Several design artifacts have been proposed to mitigate bias in ML and have been readily absorbed by practitioners (e.g. datasheets for datasets [95] and model cards for model reporting [196]). This is reflective of practices in other areas of technology development and engineering, where it is common to use a wide variety of design processes and tools to ensure that artifacts built by people are useful and align with societal values. Yet, design-based approaches to detecting and mitigating bias remain limited in the ML fairness literature, even though they have been integral to the development of AI systems [276]. By approaching the tasks of detecting and mitigating bias in ML from the perspective of design, practitioners can draw on a rich history of tools, processes and practices to interrogate how things have come to be and prescribe how they ought to be. Regarded from this lens, the shortfalls in detecting and mitigating bias discussed in the previous section can be viewed as an invitation to supplement current approaches to bias detection and mitigation in ML with design interventions in order to ensure that the technology we build enables a world that is non-discriminatory, fair, inclusive and diverse.

3.5. Design Patterns

The power of design lies not so much in novelty per say, but rather in its ability to capture abstract, recurring relationships between objects [3]. Novelty emerges by combining objects and the abstract relationships between them in a multitude of different ways, in a manner that suits a particular context. Abstraction makes it possible to move from specific design instances to repeatable approaches. A designer that has access to abstractions can draw on past experience to invoke established objects and their relationships, combining them at will to create new entities [93]. This section explores design patterns as an approach for capturing abstract relationships between objects as reusable design knowledge, accessible to designers and developers in their everyday practice. In doing this I intend to lay the foundation for using design patterns as a practice-orientated design intervention for detecting and mitigating bias in Edge AI.

3.5.1. Patterns and the Purpose of "A Pattern Language"

Patterns have a long-standing place in the history of human craftsmanship and making, from garment and shoe manufacturing, to the casting of metals and mass production of plastic items. Patterns exist to be copied, they are "a form or model proposed for imitation" [195]. In traditional making and manufacturing, patterns aid makers to reconstruct the final form, they enable repetition and reproduction. Tracing the history of patterns in software design leads back to the architect Christopher Alexander, who studied, described and defined patterns as building blocks of the architectural design process [93, 231, 246]. In their highly cited work *A Pattern Language* [6, p. x], Alexander et al. describe patterns as follows:

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

Alexander's Pattern Language

In Alexander's work, patterns were not invented, but observed from people's interactions with existing structures and the built environment. The patterns have a quality of invariance across time, and many of Alexander's patterns still ring true today. For example, as a person with a height of 158cm, living in a country where the average person is more than 20cm taller than me, pattern 251 pattern has particular appeal [6, p. 1158]:

*"**Different chairs** - people are different sizes, they sit in different ways so furnish with a variety of different chairs."*

Alexander's approach to describing and discovering patterns that capture reusable solutions to problems observed in existing systems appealed to the software engineering community. His pattern language is recognised as an influential source of inspiration in

software design, particularly object-oriented software [93, 231]. However, Alexander himself was initially oblivious to the influence his work had on the software engineering community, and he never took an active role in shaping the emerging discipline [4]. In Alexander's own work, the pattern language was not the end, but an intermediate point from which his later work, *The Nature of Order* [5] emerged. Rather than focusing on the specifics of his pattern language, I want to highlight the motivations behind the language, which Alexander shared in a keynote address at the ACM Conference on Object-Oriented Programs, Systems, Languages and Applications (OOPSLA) in 1996 [4].

Designing for Aliveness and Coherence

A driving motivation for Alexander was to use architectural design as a tool to make the environment nurturing for human beings, and to do this at a really large scale. His intention was for the patterns to have a "profound impact on human life" [4]. He held the core belief that a good environment is one that makes people come alive. To Alexander, this required local adaptation, which again required participation and agency of people who are locally knowledgeable. Thus, in his pattern language, Alexander aimed to go beyond merely identifying structural features that shape the environment to be positive and nurturing, and to (re)open the design process to participation of those who experience the environment.

Alexander identified three essential features that capture the purpose of his pattern language. Firstly, the language had a moral component, that compelled designers to create designs that nurture *aliveness*. Alexander was concerned with the question whether the patterns result in designs that make people more whole in themselves. Secondly, he wanted patterns to result in artifacts that are coherent and whole, that have structural integrity. Thirdly, he viewed the patterns as generative, wanting them to encourage and enable design processes that allow people to create what he considered to be coherent and morally sound design.

Differences, Critiques and Take-aways

In their seminal work on object-oriented software design patterns, Gamma et al. [93] acknowledge Alexander's pioneering work. However, the authors also highlight that architectural and software engineering patterns differ in many ways, ranging from the design material, to the maturity of the discipline, constraints around pattern usage, the focus on problem as opposed to solution, and the claims of what patterns, if applied well, can achieve. Furthermore, even though Alexander is frequently associated with patterns in software design, not everyone in the software engineering pattern community considers his work as influential or even relevant [184]. In the architectural domain, critics of Alexander's pattern language have found his approach to fall short on embracing pluralistic values and alternate experiences, on accounting for political and social realities, and on achieving the ideals that underlie the patterns [51].

Despite these differences and critiques, in the current climate of AI development where the bias challenges discussed in the previous sections are not rare occurrences but the norm, Alexander's intentions for creating a pattern language to produce artifacts that make people come alive, that is participatory and generative, resonate with me. I hope that by creating design patterns for detecting and mitigating bias in Edge AI, I can offer a tool for developers to contemplate the moral impact of their designs, to conduct bias evaluations that are coherent, and to develop Edge AI applications that are non-discriminatory and inclusive.

3.5.2. Design Patterns in Software Engineering

Design patterns are widely used to capture design knowledge and experience in diverse areas of software engineering. For illustrative purposes, I highlight a few examples. Patterns have been developed for GIS applications [99] and e-learning systems [244], for dynamically adaptive [205], safety-critical [9] and cyber-physical systems [205], and on a higher level for process modelling of workflow management systems [302], organisational data systems [111], and data quality management [309]. The concept of patterns is too commonly and too widely used to ascribe it to a single discipline, let alone a group of people or even an individual. Yet, there have been some works on patterns in software engineering that have been particularly influential. Before reviewing the perspectives on patterns in these works, there is one important distinction that is necessary to make. Patterns are well known in machine learning, where there is an entire community that has developed techniques for pattern recognition from data [22]. When speaking about patterns in this chapter, I refer to patterns in the sense that Christopher Alexander would have used the concept, not in the sense that corresponds with the work of the machine learning and pattern recognition community.

The GOF Design Patterns in Object-Oriented Software

The object-oriented software design patterns of Gamma, Helm, Johnson and Vlissides, commonly referred to as the "Gang of Four" (or just GOF), are perhaps the best known example of patterns used in software design [93]. The GOF viewed design patterns as an intervention to a common software engineering problem: that software engineers often fell short of recording their software design experience for others (and themselves) to reuse. They proposed design patterns as a tool to record "*a solution to a problem in a context*" [93, p. 3]. The four essential elements of a GOF design pattern are its name, the problem it addresses, the solution it describes and the consequences that result from using the pattern.

The GOF collected their patterns in a pattern catalogue, with the goal of capturing important and recurring designs in a manner that can be easily understood by engineers. Their pattern catalogue is inherently developer-centric to help developers of new systems "get a design 'right' faster" [93, p. 2]. With the catalogue, the GOF wanted

to make proven software design techniques accessible and reusable. Added benefits of this were that patterns could help engineers choose between design alternatives, provide explicit specifications of component interactions, and clarify the underlying intent behind their design choices. This again could improve documentation and system maintenance.

Patterns Beyond Design

A contemporary of the GOF, Peter Coad [231] also studied object-oriented patterns. He considered them useful as building blocks that standardise "small piecework" [231, p. 152] in the analysis and design of object-oriented development. To Coad, patterns represented relationships between the lowest-level elements in object-oriented development, "*an abstraction of a doublet, triplet, or other small grouping of classes that is likely to be helpful again and again...*" [231, p. 153].

Like Coad, Riehle and Zullighoven [246] considered patterns to be useful not only in design, but at different stages in the software development process. They conceptualised a pattern broadly as "*the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts.*" They aligned their conceptual, design and programming patterns with established software engineering modeling approaches for application domains, software design and implementation. Conceptual patterns are closely connected to the application domain, design patterns are described by means of software design constructs, and programming patterns are described with programming language constructs. Riehle and Zullighoven note that patterns are more than guidelines for software development. They provide a common language to communicate about shared concepts, and become part of the culture of a team or community.

The distinction between analysis and design is however not always that clear. Instead of focusing on the boundary between the two, Martin Fowler [184] considered it important that the structure of a developed artifact reflects the structure of the problem that it solves. He used conceptual models as a tool to understand and simplify problems, and proposed patterns as a tool in conceptual modelling. Fowler's definition of a pattern is "*an idea that has been useful in one practical context and will probably be useful in others*" [184, p. 8]. This perspective is purposefully broad to highlight that a pattern can be anything. Fowler defined analysis patterns as "*groups of concepts that represent a common construction in business modeling. It may be relevant to only one domain, or it may span many domains*" [184, p. 8]. Additionally, he identified supporting patterns that "*describe how to take the analysis patterns and apply them*" [184, p. 8]. Like Riehle and Zullighoven, Fowler saw a major advantage in the ability of patterns to construct a common language to facilitate communication and project collaboration.

Commonalities in Pattern Purposes

Despite defining patterns at different levels of abstraction and for different stages in software development, there are three common themes expressed by these defini-

tions that underlie the purpose of patterns in software engineering: utilitiy, context-specificity and reusability. First, is the idea of *utility*, expressed by the GOF as a "solution to a problem", by Coad as "helpful" and by Fowler as "useful". These expressions emphasise that patterns are an artifact with a purpose, and are only good in so far as they achieve that purpose. Second is the need for patterns to be *context specific*. The GOF, Riehle and Zullighoven, and Fowler explicitly speak of practical, specific non-abstract contexts to which patterns apply. This emphasis is understandable when viewed from the perspective of design, where context specificity is of uttermost importance to locate artifacts in a place, over a period of time, amongst a group of stakeholders. Dislocating patterns and context erodes the utility of patterns. The third theme that emerges is that of *reusability*. Coad requires patterns to be helpful "again and again", Riehle and Zullighoven speak of "recurring" abstractions and Fowler of ideas that can be transferred from one context to be useful "in others". The need for reusability echoes the purpose that patterns for making have carried through the ages: patterns capture and represent experience in an abstract form for imitation to enable predictable outcomes and reduce resource investments required for reproduction. A community that can capture their joint experience in accessible patterns has a tool for collective learning, as they can formulate and communicate how to repeatedly reproduce their work.

3.5.3. Identifying, Validating and Formulating Patterns

From Practice to Patterns to Practice

Design patterns are strongly rooted in practice and the experience of creating existing artifacts. As Fowler put it, "patterns are discovered rather than invented" [184, p. 8]. Discovery, according to Fowler, comes from observing routine software development. Fowler's patterns were developed from his personal, practical experience and highlight aspects of projects that he did. Coad expressed a comparable view some years earlier, stating that patterns can be observed or found by trial-and-error [231]. The GOF left the question of how their patterns were identified and validated similarly open-ended, requiring only that patterns must come from designs that had been applied more than once, in different systems [93]. In capturing the designs as patterns, their contribution was one of documenting best-practice knowledge for the first time. Some later works have looked at discovering design patterns with automated means like machine learning [86]. However, automated approaches appear to be more frequently used to reverse engineer patterns from existing code [68, 232, 265] rather than for finding new patterns without any patterns to start with.

Patterns come from practice and are created for practice. Riehle and Zullighoven [246] categorise patterns as having a generative or a descriptive function in software development practice. Generative patterns, like those of Alexander's pattern language, are primarily intended to be used to derive new artifacts. Descriptive patterns, as they

consider the GOF pattern catalogue, place more emphasis on describing pattern attributes and how to use them. However, the function of a pattern is not definitive, as patterns can be rewritten to be more generative or more descriptive. Patterns can be implemented as concrete instances, and used to recognise pattern instances in existing systems. Oftentimes, patterns relate to each other, either through interaction or embedding. Patterns can also be grouped into collections. Such pattern sets may have structure, like ordering or hierarchy.

Pattern Forms and Templates

Pattern formats and opinions on formats vary between pattern creators. Broadly, there are those that use a fixed pattern format, most notably the GOF, and others, like Fowler, Riehle and Zullighoven who consider a less rigid approach more suitable. There is one aspect of form that is commonly used by all pattern creators, and that is the pattern name. Names allow practitioners to concisely refer to the vast design experience that a pattern represents. By naming patterns, they offer practitioners a shared vocabulary founded on common understanding [184].

The fixed format patterns typically follow the form of either Alexander or of the GOF [246]. Patterns in Alexander's form consist of at least three sections to describe 1) the problem, 2) the context and forces from which the problem arises, and 3) the solution that the pattern presents. The GOF template is tailored to object-oriented design. The GOF introduced the template to give patterns a uniform structure, which they believed would make them easier to learn, compare and use [93]. While not explicitly modelled on Alexander's patterns, their "Intent" section corresponds with Alexander's problem section, "Motivation", "Applicability" and "Consequences" are concerned with context, and the remaining sections predominantly describe the solution and how to apply it.

Aligned with their view that the best description for a pattern depends on its intended use, Riehle and Zullighoven proposed a general pattern form that consists only of a context section and the pattern itself [246]. They found that this descriptive form gave their patterns flexibility to be used for multiple purposes, for example to create pattern instances, to find and recognise instances and to compare patterns. To overcome the limitations that this flexible format presents for specific applications, they supplemented the form with additional sections that can be adapted for particular application. Fowler did not find that the commonly used fixed format templates, or even their elementary sections like problem-solution pairs, were always suitable for the patterns he wanted to capture. Problems could be solved in more than one way, and often depended on trade-offs. Discussing several solutions together instead of breaking them down into individual patterns thus has its own advantages, especially for conceptual modelling. Nonetheless, Fowler considered it important that patterns are not too complex, that they are widely applicable across a large domain and that they are based on an effective conceptual model of that domain.

3.5.4. Design Patterns in Artificial Intelligence

Design patterns are gaining traction in ML and deep learning. Washizaki et al. [318] present 15 software-engineering design patterns for machine learning applications that cover the overall system topology and architecture, programming aspects related to the design of a particular component, and operational aspects that focus on ML model and prediction quality. While the authors suggest that design patterns for model and prediction quality can address aspects of robustness, explainability, prediction accuracy and fairness, they do not propose design patterns that specifically address bias or fairness related concerns. Take et al. [279] adapt design patterns from the GOF, such as *factory method* and *strategy*, to the AI context but do not adapt them to trustworthy AI or bias and fairness concerns. Tuggener et al. [297] focus on a particular application context and propose design patterns for resource constrained automated deep learning methods. Outside of the academic literature, Lakshmanan et al. [161] have published the book "Machine Learning Design Patterns" for developers. Their patterns include three responsible AI patterns, one of which addresses bias and fairness related aspects. However, given the complexity of considerations that must be taken into account when detecting and mitigating bias in ML, this pattern conveys the importance of the topic but offers limited guidance on what to do about it.

Several researchers have proposed responsible AI design patterns. Elish and Hwang [75] present an AI pattern language for responsible AI. However, their patterns are formulated on a very high level and read as aspirations rather than as best practice knowledge that can readily be implemented in new contexts. Dzambic et al. [71] propose two architectural patterns for integrating AI technology into safety-critical systems, but do not focus on bias and fairness. Lu et al. [177] formulate software engineering patterns to operationalise high level AI ethics guidelines into responsible design practice. While they include several patterns that relate to ethics, risk assessment and testing, they do not capture design practices that specifically relate to bias and fairness. These nascent efforts underscore the potential of design patterns for detecting and mitigating bias in ML more broadly and Edge AI in particular. Yet, to date, the development of such patterns remains a gap in the literature.

3.6. Conclusion

This chapter reviewed related work on trustworthy AI and IoT, bias in ML, approaches and shortcomings of detecting and mitigating bias, and design patterns. The chapter observed that trustworthiness perspectives in the AI and IoT domains do not readily align, and proposed a unified perspective for trustworthy Edge AI that considers both conceptualisations. Based on this joint perspective, the chapter then identified diversity, non-discrimination and fairness as an attribute of trustworthy Edge AI that has not been studied in the literature. Next the chapter, clarified key concepts of fairness, discrimination and bias, and examined bias as a source of unfairness and discrimina-

tion in ML. This section contributed a diagrammatic representation of bias feedback loops in AI systems to visualise how cognitive biases, design choices and data inputs contribute to bias in AI systems.

The chapter proceeded to review literature on bias detection and mitigation in ML, covering approaches that quantify bias and highlighting their shortfalls. An overview was provided on literature that points to implementation challenges that practitioners experience when trying to detect and mitigate bias in practice. These challenges pertain to limitations of current tools for bias detection and mitigation, including inconsistent and poorly documented methodologies, limited coverage and an oversimplification of bias concerns. They also highlight that practitioners lack a language to communicate about bias in interdisciplinary teams, that many practitioners have limited background knowledge on bias detection and mitigation and that learning about best practices for bias detection and mitigation is seldom supported. The challenges are indicative of gaps in the literature, which thus far has focused primarily on algorithmic fairness techniques. The chapter notes that only few studies have developed design artifacts and that these have been readily absorbed by practitioners. The development of design interventions for bias detection and mitigation is thus identified as a gap in the literature.

This thesis views the development of design patterns as a promising intervention for addressing the observed bias detection and mitigation challenges and gaps, and ultimately for enabling bias detection and mitigation in Edge AI in practice. The chapter thus introduced design patterns as a generalisable approach for detecting and mitigating bias in Edge AI that can overcome implementation challenges currently experienced by practitioners. Design patterns are first reviewed from the perspective proposed by the architect Christopher Alexander. His aspiration of using design patterns to design for aliveness and coherence, to open design to participation and to contemplate the impact of design choices remain an inspiration to this thesis. Next, design patterns in software engineering and approaches to identifying, validating and formulating them were reviewed. These approaches strongly influenced the patterns developed in this thesis. Finally, several design patterns have recently been proposed for AI and even trustworthy AI. A review of this research indicates that no prior studies have developed design patterns that can be used to detect and mitigate bias in ML workflows. The next chapter builds on the work reviewed here to identify design patterns for detecting and mitigating bias in Edge AI.

4

Identifying Patterns for Detecting and Mitigating Bias in Machine Learning

Chapter 3 established the need for, and challenges in developing trustworthy Edge AI, highlighting bias detection and mitigation as a particular challenge area. Despite advances made to detect and mitigate bias in machine learning, current approaches are not easily conveyed to Edge AI developers. However, design patterns are a mature tool in software engineering that can overcome many of the practical and implementation challenges that practitioners face when trying to detect and mitigate bias. This chapter sets out to find design patterns to detect and mitigate bias in machine learning. The emphasis lies on finding patterns from established knowledge and practice, not on inventing new patterns. The chapter proposes a conceptual model for guiding pattern identification. It then identifies pattern instances in software tools, consolidates them into patterns and organises the patterns in a catalogue. Finally, the chapter formulates two recipes for using a collection of patterns to measure bias and curate benchmark datasets.

4.1. Introduction

This chapter initiates the design science research cycle by identifying, categorising and describing design patterns. While the ultimate objective of this thesis is to formulate design patterns for detecting and mitigating bias in Edge AI, research on bias detection and mitigation in Edge AI is scarce as pointed out in Section 3.2.4. This chapter thus identifies established practices for bias detection and mitigation from machine learning (ML) fairness, and uses these to formulate design patterns.

The literature review in Sections 3.3.2 highlighted that sources of bias in ML often interact in complex, non-linear and dynamic ways. This makes the practical implementation of bias detection and mitigation in ML development workflows a challenging undertaking. Practitioners use fairness toolkits to assist them, but as described in Section 3.4.2, current toolkits fall short in several regards. They lack consistency, are difficult to compare and require a high degree of expertise in ML fairness from practitioners [166]. Yet, most practitioners have little or no training in ML fairness, and consequently use toolkits in inappropriate ways [56]. Practitioners find it difficult to convey their results to others, which limits team collaboration and effective motivation of fairness concerns to organisational stakeholders.

Section 3.5 introduced design patterns as a mature tool in software engineering, that is gaining traction in ML and deep learning [161, 279, 297, 318] and responsible AI [71, 75, 177]. In software engineering design patterns are practitioner oriented and capture established experience in a domain [93]. They formulate recurring concepts in an abstract way so that the concepts can be readily used in new designs. Through their emphasis on reusability, design patterns promote access to and transfer of knowledge [231, 246]. By using design patterns, practitioners that are new to a domain can quickly familiarise themselves with important concepts in the domain. In practice, design patterns have been found useful to establish a shared vocabulary and help stakeholders reach a common understanding of complex concepts, thus aiding communication [184]. Design patterns have not been used to detect and mitigate bias in ML, but they could address many of the challenges encountered by practitioners.

This chapter finds design patterns for detecting and mitigating bias from established practices in ML fairness that have been programmed and documented in software tools. The chapter starts by defining patterns, describing their purpose and affordances, and listing requirements in Section 4.2. Section 4.3 describes the approach for identifying patterns from software tools that detect and mitigate bias. The patterns are presented and organised in a catalogue in Section 4.4, and their instantiation in software tools is discussed. In Section 4.5 two pattern recipes, which are purposeful collections of patterns, are proposed for guiding bias measurement and benchmark dataset processes. The patterns are validated against requirements and the limitations of the approach are discussed in Section 4.6. Finally, Section 4.7 summarises the contributions of this chapter.

4.2. Pattern Definition and Requirements

This section describes the purpose of patterns, defines them and describes their affordances in the context of this thesis, and concludes with a list of pattern requirements. The section draws on prior literature on patterns in software engineering, which has been discussed in detail in Sections 3.5.2 and 3.5.3.

4.2.1. Pattern Purpose

The patterns developed in this thesis should enable practitioners to detect and mitigate bias when developing Edge AI applications, and serve the end goal of ensuring that Edge AI applications are non-discriminatory and inclusive. The patterns are thus purposefully practitioner-centric. Practitioners, as described in Section 1.2, are people who have technical competency in developing or overseeing the development of an aspect of the Edge AI development workflow (e.g. researchers, developers or product managers), but have no or very limited prior experience in detecting and mitigating bias in ML. They are likely to interface with non-technical stakeholders in their organisation, and will need to communicate bias investigations and interventions to them. To use the patterns, practitioners must have some familiarity with the concepts of bias, fairness and discrimination introduced in Section 3.3.1, and a willingness to interrogate bias.

4.2.2. Pattern Definition

Several different pattern definitions have been proposed in the literature (see Section 3.5.2), with the GOF definition of a pattern [93] as *a solution to a problem in a context* being frequently referenced in software engineering. On a high level, bias could be viewed as a problem, and bias detection and mitigation could be viewed as solutions. However, in reality the distinction between problems and solutions captured by patterns for detecting and mitigating bias may not always be clear. By contrast, Martin Fowler [184] defines a pattern as *an idea that has been useful in one context and will probably be useful in others*. This definition is more general and emphasises utility, reusability and context-specificity as three important aspects of patterns. Building on Fowler's definition, this thesis adopts the following definition:

A pattern for detecting or mitigating bias in ML is an idea that has been found useful or necessary for detecting or mitigating bias in one practical context, and is likely to continue being useful or necessary for that purpose in other contexts.

From this definition it follows that a pattern must be *useful* for the purpose of detecting and mitigating bias. A pattern must be *reusable*, so that practitioners that use it benefit by accessing collective experience that helps them to formulate and communicate their work, and reduces the resources required to do their work. Finally, a pattern must be

context-specific, which is important as design, bias, fairness and discrimination are also context specific.

4.2.3. Pattern Affordances

Just as pattern definitions vary in the literature, so do pattern affordances. The affordances discussed here are inspired by the affordances observed in the literature in Section 3.5, and highlight the potential of patterns to overcome many of the challenges that practitioners experience when trying to detect and mitigate bias in ML [56, 166]. Patterns should capture a complex subject as simply as possible, without risking oversimplification or loosing contextual nuance [184]. Patterns should help practitioners interrogate the impacts of their designs and conduct bias evaluations that are coherent, reproducible and consistent across practitioners and time. Individual patterns may be meaningful on their own, or in relation to other patterns [231]. Experts should see their domain reflected in patterns and only rarely be surprised by them. Novices should be able to gain a more complete understanding of a domain with every pattern that they master.

Patterns as building blocks

In the literature, patterns have been developed to capture different levels of knowledge, ranging from complex, multi-step processes [309] to the lowest-level elements that standardise building blocks of larger designs [231]. This thesis perceives patterns as building blocks, and combines patterns into recipes (described later in Section 4.5) to capture processes. It is thus possible that a single pattern which represents a solution to a particular problem relating to bias is insufficient to detect or mitigate bias on its own. Nonetheless, it may be a recurring concept that is not only useful but essential for detecting or mitigating bias in a variety of contexts.

For example, the equalised odds ratio, which is a type of algorithmic fairness measure, is too specific to be a pattern as it captures an instance of an idea that can be expressed in a more general form. The more general form is expressed by its parent category, the algorithmic fairness measure concept, which captures the idea of quantifying performance differences in algorithmic systems across groups of people. In this case the algorithmic fairness measure is a pattern[1], representing a building block for bias detection and mitigation, while the equalised odds ratio is an instance of the pattern. By contrast, the process of measuring bias involves not only the quantification of performance differences, but also multiple other steps, like deciding how to group people. In this thesis entire processes, like measuring bias, are considered to be too complex to be a pattern as they can be broken down into smaller building blocks. Conversely, several patterns can be combined in a collection to describe a process [231, 246], such as an evaluation protocol for measuring bias.

[1]In the pattern catalogue in Section 4.4 it is listed as the *Bias measure* pattern

Patterns to generate and describe

As discussed in Section 3.5.3, patterns can have a generative and a descriptive function [246]. This research anticipates that patterns will take on both roles. In their generative role, patterns may be used to derive new artifacts. For example, patterns may articulate how to measure bias in a particular domain, or formulate a procedure for mitigating bias in a specific type of application. In their descriptive function, pattern attribute and usage descriptions can offer practitioners a tool for learning and deliberation. Irrespective of whether patterns are used to generate or describe, they should establish a common vocabulary, which should help teams reach consensus on important concepts, and facilitate conversations and collaboration between interdisciplinary stakeholders.

Pattern implementations

Patterns can be implemented as concrete instances [246], such as classes or methods in software that detects and mitigates bias. For example, if a practitioner has written a programme to calculate the equalised odds ratio, the programme is an implementation of an instance of a pattern. A prospective use of the patterns is to recognise pattern instances that are implemented or documented in existing systems. This could make software more comparable, and could enable auditing of systems through reverse engineering.

4.2.4. Pattern Requirements

The pattern purpose, affordances and corollaries that follow from the definition can be consolidated into a list of requirements that describe the desired pattern functions, content, interaction and their form. The requirements have been created to serve as inclusion and exclusion criteria when identifying patterns, and to provide guidance on how to formulate patterns. While grouping pattern requirements by function, content, interaction and form was inspired by the work of Riehle and Zullighoven [246], and while many of the requirements are reflective of statements about patterns from the literature in Section 3.5, the requirements are specific to the context of bias detection and mitigation in ML. They are listed below.

Pattern Form

1. A pattern must be unique.
2. A pattern must have a name.
3. A pattern must express an idea that supports bias detection or mitigation in ML.
4. A pattern must be formulated as simply as possible, without risking oversimplification or loosing contextual nuance.
5. A pattern must be accessible to practitioners with limited experience in bias detection and mitigation in ML.

Pattern Function

6. A pattern must support practitioners in detecting or mitigating bias in ML.
7. Patterns, if used together, must enable bias evaluations and interventions that are coherent, reproducible and consistent across practitioners and time.
8. Patterns, if used together, must help practitioners interrogate the impacts of their designs.
9. A pattern must promote reusability, so that practitioners that use it benefit from collective experience.

Pattern Content

10. A pattern must have been found useful or necessary for detecting or mitigating bias in one or more practical contexts.
11. A pattern must be adaptable to different application contexts.
12. A pattern must be applicable to a stage of the ML development workflow.
13. A pattern is not an instantiation of a particular method and not specific to a programming language.

Pattern Interaction

14. A pattern must be meaningful on its own, or when used with other patterns.
15. A pattern must be consistent with other patterns.
16. A pattern does not have to detect or mitigate bias on its own, but must then contribute to a collection of patterns that can do this.
17. A pattern collection must achieve a particular bias detection or mitigation objective.

4.3. Approach for Identifying Patterns

There exist many ways of identifying patterns and many ways of synthesising them into a pattern catalogue. This thesis treated pattern identification as an iterative design activity through which raw ideas were refined into well articulated formulations. This section describes the approach that was taken to identify patterns and develop the pattern catalogue. The approach is not a prescriptive method, but rather a record of analysis and design decisions, captured for reproducibility, transparency and to make assumptions explicit. The patterns that are derived through this approach should be regarded as a first iteration of the patterns, and not as their final form. The section first presents an overview of the approach, then introduces a conceptual model for categorising patterns, proposes how patterns can be used in the ML development workflow and lastly describes a reverse engineering approach for systematically identifying patterns implemented in software tools that detect and mitigate bias.

4.3.1. Overview of Approach

Like many pattern creators before me, I hold the view that patterns should come from practice and be useful in practice [93, 184, 231]. Throughout this thesis I have been engaged as a practitioner in projects involving bias evaluation in ML and responsible AI design practices. The idea of patterns first emerged out of a practical need I experienced while trying to convey ML bias and fairness concepts to project collaborators who had limited prior experience in this domain. The first projects started in 2020, and some are still ongoing. They are listed below:

1. The Zen of ML Project[2] with the Mozilla Trustworthy AI Working Group
2. Bias Tests for Voice Technologies (bt4vt[3]) python software library
3. Research collaboration with speech researchers in academia, which underpins Section 5.5
4. Research collaboration with researchers in on-device ML and ubiquitous computing in industry, which underpins Chapter 6
5. Book review and breakout on responsible design and AI ethics in Situnayake and Plunkett [272] "AI at the Edge"

Once the projects were underway, I observed recurring questions about bias detection and mitigation from collaborators with distinctly different backgrounds. Having experienced many of their challenges myself when I first started studying bias in ML, this was a strong motivation to find a generalisable approach to address their challenges in order to facilitate bias detection and mitigation. An overview of the approach that ensued is captured in Figure 4.1. In following the approach, three artifacts were developed: a conceptual model, a pattern catalogue, and pattern recipes.

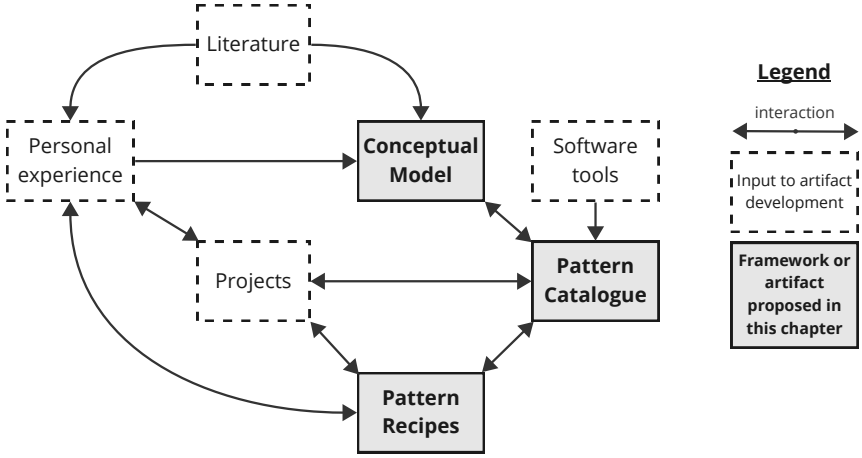


Figure 4.1: Overview of pattern identification approach

[2] https://wiki.mozilla.org/Working_Groups/Zen_of_ML
[3] https://github.com/wiebket/bt4vt

The conceptual model was developed first to provide a top-down framework that can categorise patterns and guide pattern discovery. It was influenced by literature and by personal experience that I acquired while leading or collaborating with others on the aforementioned projects. The conceptual model is described in Section 4.3.2. The pattern catalogue was developed next, and is a collection of all the patterns that have been identified from software tools, using the categorisation provided by the conceptual model. The bottom-up process for finding patterns from software tools is described in Section 4.3.4 and the pattern catalogue is presented in Section 4.4. The pattern recipes were developed last to demonstrate how patterns from the catalogue can be used in practice. They are purposeful collections of patterns that guide specific processes necessary for bias detection or mitigation. The recipes presented in this chapter were influenced by my personal experience, which again was shaped by best practices for bias measurement and benchmark datasets that I observed over time in literature and projects. The recipes are described in Section 4.5.

Many of the insights about the patterns were gained while developing the projects, and it is at times difficult to tell which came first, the patterns or the practice. This dual approach turned out to be an important aspect of the pattern development process, and is represented by the bi-directional arrows in Figure 4.1 that indicate mutual interaction between the artifacts and their inputs. For example, the pattern catalogue was used to create the pattern recipes, but later the recipes were used in projects, and patterns were added to expand the recipes where necessary. The projects thus evolved the recipes, which yet again expanded the pattern catalogue. Despite the circularity during development, this chapter captures a linear narrative that illustrates how the conceptual model was used to identify patterns which were used to create recipes. The reverse interactions with projects will become visible in Chapters 5 and 6.

4.3.2. Conceptual Model for Pattern Identification

To guide the pattern discovery process, this chapter proposes a conceptual model in Figure 4.2 to categorise patterns. Inspired by the classification system proposed by Gamma et al. [93], patterns are categorised on a high level by their purpose and scope.

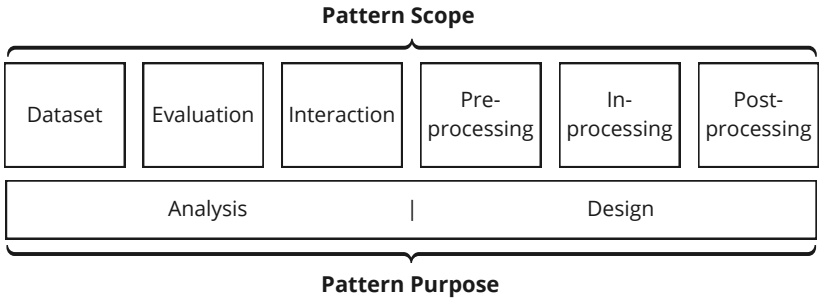


Figure 4.2: Conceptual model of high-level pattern categories

Pattern Purpose

Conceptually, the model likens bias detection and mitigation to analysis and design activities in the development of ML artifacts. Patterns can thus be categorised broadly as serving a design or an analysis purpose. Analysis patterns capture transferable knowledge for detecting sources of bias, offering insights that can help practitioners identify bias. They typically do not offer guidance on interventions to reduce bias, which remains the role of design patterns. Design patterns capture transferable knowledge for bias mitigation.

Pattern Scope

The ML worfklow (see Figure 2.2) offers an intuitive way to guide practitioners on where to expect sources of bias in the artifact development process. The algorithmic fairness literature frequently classifies bias mitigation strategies as pre-processing, in-processing and post-processing interventions, based on where they are applied in the ML workflow [16, 192]. The conceptual model draws inspiration from this to classify the scope of patterns. Dataset patterns correspond with the data gathering stage of the ML workflow, while evaluation patterns correspond with model testing, model deployment and ongoing monitoring. Pre-processing patterns relate to data processing. In-processing and post-processing patterns relate to model training. Interaction patterns are not specific to a particular stage of the ML workflow, but rather present approaches to interrogate and interact with other patterns and with the ML system at different stages of the workflow.

In my experience, patterns with a dataset, evaluation or interaction scope tend to be analysis patterns, while patterns with a pre-, in- or post-processing scope tend to be design patterns. Dataset patterns frequently describe and analyse datasets used in ML systems in order to detect sources of bias that can arise during data collection and dataset curation. However, dataset patterns could also transfer prescriptive design knowledge on how to construct datasets that limit bias. Such patterns would then be design patterns. As the nature of evaluation is analytical, evaluation patterns are very likely to be analysis patterns. Similarly, interaction patterns typically do not offer insights on mitigating bias, but help practitioners to detect bias, making them more likely to serve analysis purposes. Pre-processing, in-processing and post-processing patterns offer approaches to mitigating bias, and are thus more commonly associated with design patterns. While common, these associations are not fixed and patterns of any scope could have either purpose.

4.3.3. Using Patterns in the ML Development Workflow

Bias detection and mitigation are interdependent actions. It is not possible to mitigate bias without being able to detect it. Similarly, if sources of bias are detected but no mitigating actions are taken, then biased ML systems remain at risk of being discriminatory and exclusionary. Just as bias detection and mitigation are interdependent, so

are design and analysis patterns. While design patterns can mitigate bias, they co-exist alongside analysis patterns, which are ultimately necessary to assess if interventions for bias mitigation successfully reduced bias. This iterative application of analysis and design patterns is well suited to the iterative ML development process.

The purpose and scope of patterns provides guidance on where and how patterns can be used in the ML development workflow. The pattern scope suggests whether a pattern should be used during data gathering, model development or evaluation. The pattern purpose suggests whether a pattern should be used for bias detection or mitigation. Analysis patterns can be used to detect bias in a ML system, but they will rarely be used on a once-off, standalone basis. If bias is detected with the help of analysis patterns, then design patterns offer interventions to mitigate bias. In applying design patterns, the ML system is revised and updated from its initial version. However, to confirm if bias has indeed been mitigated, analysis patterns are once again necessary to detect bias. This interplay and co-dependency of analysis and design patterns for detecting and mitigating bias is visualised in Figure 4.3.

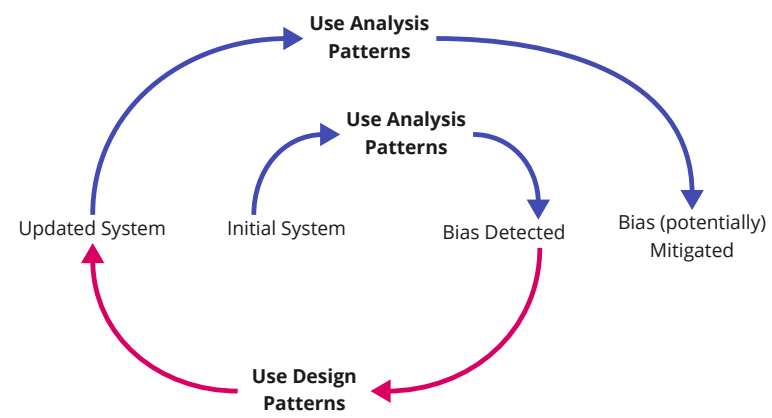


Figure 4.3: Interaction between analysis and design patterns

4.3.4. Reverse Engineering Patterns from Software Tools

Granular patterns that can assist practitioners with detecting and mitigating bias have not been studied in the literature. To identify patterns, this research thus turns to software tools that have been developed for the purpose of bias detection or mitigation. The programmed methods implemented in these software tools and the concepts described in accompanying user documentation can be viewed as potential instantiations of implicit patterns that have been found useful or necessary in practice. The challenge then is to identify these functions and concepts, to consolidate them across tools, and to formulate them explicitly as patterns. This section describes the approach followed in this thesis to reverse engineer patterns through a bottom-up analysis of software tools.

These tools operationalise knowledge and practices that tool creators deem useful or necessary for bias detection and mitigation. The approach thus ensures that patterns that are identified have been found useful or necessary for detecting or mitigating bias in one or more practical contexts, and meet Requirement 10.

Reverse Engineering Approach

The steps involved in the reverse engineering process are visualised in Figure 4.4. First, software tools were selected for analysis from the literature and practical experience. Once the tools were selected, their source code and documentation were analysed to identify programmed methods and fairness related concepts which could be categorised as potential pattern instantiations. The conceptual model was used to provide top-down guidance during this process and to help categorise pattern instantiations. In total, 106 pattern instances were identified, captured and categorised. I then consolidated similar pattern instantiations into unique patterns, once again taking guidance from the conceptual model and from personal experience. As the final step, the patterns were gathered in a catalogue.

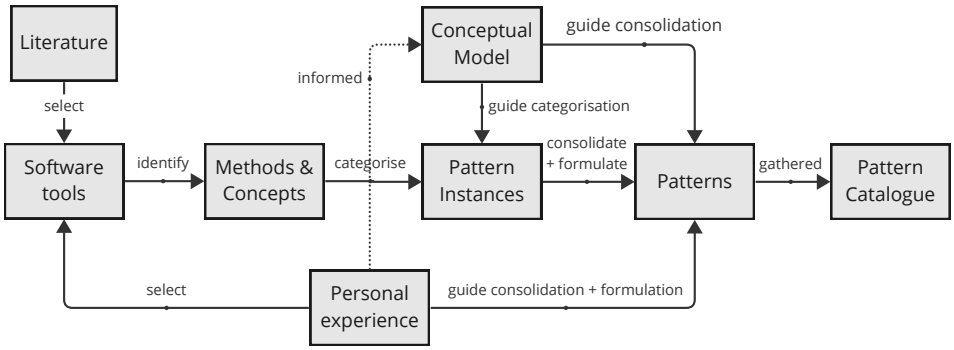


Figure 4.4: Reverse engineering approach for identifying patterns from software tools by analysing programmed methods in source code and fairness concepts in user documentation.

Formulating Patterns

The patterns were formulated as *name/key idea* pairs. As discussed in Section 3.5.3, this form offers greater flexibility than fixed format templates [184]. Given that the patterns identified in this chapter present a first iteration and not the final artifact, this flexibility is an advantage. It provides an opportunity to observe if forms emerge repeatedly when using the patterns in use cases in Chapters 5 and 6, and leaves it open that different patterns require different templates to capture actionable steps for practitioners. It is thus possible that a more extensive form will evolve as the patterns mature. Nonetheless, the *name/key idea* pairs ensure that patterns meet Requirements 2 and 3.

Selection of Software Tools

To select the software tools for analysis, a table of toolkits for bias detection and mitigation was compiled from multiple sources. First, the six toolkits analsysed in the toolkit survey of Lee and Singh [166] were selected. Next, a literature search was conducted on Scopus to look for papers that matched the query "*(bias OR fairness) AND (machine+learning OR algorithm OR algorithmic) AND toolkit*" in the title, abstract or keywords, published in computer science, social sciences or decision sciences. The search returned 35 documents. After a manual review of the results, seven papers were selected that introduced new toolkits that were not included in [166]. Finally, I added six more tools that I encountered in practice, but that did not appear in the literature search.

In total 19 tools were considered for analysis and are listed in Table 4.1. For each tool several metadata fields were captured. The tool version, its status and Github stars (G-*) were determined on 19 December 2022. A tool was granted an active open source (OS) status if it had a source code commit to its main branch within the preceding 12 months. Next, several exclusion criteria were applied. Firstly, eight tools that were inactive, unavailable or commercial were excluded from the analysis. One toolkit (Fairkit-learn) was excluded because it used the bias detection and mitigation functionalities of the AI Fairness 360 toolkit. Of the remaining ten toolkits five were selected for further analysis. Preference was given to tools that focus on bias or fairness only, thus excluding Google's toolkits, scikit-lego and dalex. Bias scan, a very new toolkit, was also excluded as the low number of stars on Github indicates that it does not yet have an active user base.

Examples of Pattern Instantiations

The 106 pattern instantiations from which the patterns were consolidated are accessible online[4]. Table 4.2 shows examples of several pattern instantiations from this directory to illustrate how they have been implemented, how they compare across software tools and how they have been consolidated into patterns. Typically, the same main idea (e.g. "set groups") is implemented differently in different tools. For example, the Fairness Indicators tool sets groups by allowing a practitioner to define columns by which the dataset will be sliced. The idea of "intersectional groups" is then supported by specifying a list of columns in the slicer object. The AI Fairness 360 (AIF.360) tool, on the other hand, requires that a practitioner sets groups by explicitly defining privileged and unprivileged groups and implements the "intersectional groups" idea as a separate programmed method. These different implementations share the same main ideas and can be consolidated in a unique pattern called *Group design*, which is concerned with the attributes that define groups.

It is also possible that different main ideas can be consolidated into a unique pattern. For example, FairLearn implements the "maximum absolute difference" idea as

[4]All the analysed pattern instances are accessible online: https://airtable.com/shrOsrTuvM1WXKAwy

Tool Name	Source	Project Owner	Date	Status	G-*	Input
Aequitas 0.42.0 [253]	[166]	U Chicago Center for Data Science & Public Policy	2018	active OS	512	**Yes**
AI Fairness 360 0.5.0 [18]	[166]	IBM	2018	active OS	1920	**Yes**
Evaluate 0.4.0[1]	practice	Hugging Face	2022	active OS	1040	**Yes**
FairLearn 0.8.0[2]	[166]	Microsoft	2018	active OS	1433	**Yes**
Fairness Indicators 0.43.0[3]	practice	Tensorflow	2020	active OS	285	**Yes**
BeFair [33]	Scopus	Castelnovo et. al.	2020	unavailable	NA	No
bias scan[4]	practice	Algorithm Audit	2022	active OS	3	No
Certifai [115]	Scopus	CognitiveScale	2020	commercial	NA	No
CLAIMED [148]	Scopus	Elyra Project	2021	unavailable	NA	No
dalex [15]	Scopus	Warsaw U. of T. MI2AI lab	2018	active OS	1910	No
Fairkit-learn [136]	Scopus	Johnson & Brun	2019	active OS	9	No
Learning Interpretability Tool [283]	practice	Google	2020	active OS	3043	No
LiFT [304]	Scopus	LinkedIN	2019	inactive	164	No
scikit-lego[5]	[166]	Brouns, Warmerdam & others	2019	active OS	936	No
Silva [326]	Scopus	Cornell Univesity	2020	unavailable	NA	No
What-if Tool [319]	[166]	Google	2018	active OS	765	No
audit-ai[6]	[166]	Pymetrics	2018	inactive	288	No
FairML[7]	practice	Julius Adebayo	2016	inactive	338	No
Know-Your Data[8]	practice	Google	2021	inactive	237	No

Table 4.1: List of software tools that have capabilities for detecting or mitigating bias. **G-*** is the number of Github stars, **Input** denotes whether tools were analysed for patterns.

[1] https://huggingface.co/docs/evaluate/index
[2] https://fairlearn.org/
[3] https://www.tensorflow.org/responsible_ai/fairness_indicators/guide
[4] https://github.com/NGO-Algorithm-Audit/Bias_scan
[5] https://github.com/koaning/scikit-lego
[6] https://github.com/pymetrics/audit-ai
[7] https://github.com/adebayoj/fairml
[8] https://knowyourdata.withgoogle.com/

a programmed method that first computes the difference of base metrics between all pairs of groups, or between all groups and the overall base metric. The function then selects the maximum absolute value of the differences and returns it as an overall measure of model bias. Aequitas, by contrast, implements a “get unbiased decision” idea

Tool name	Instantiation	Main idea	Scope	Pattern name
Fairness Indicators	*tfma.slicer.SingleSliceSpec(columns)*	Set groups, Intersectional groups	Evaluation	Group design
AIF.360	*aif360.metrics.DatasetMetric(..., privileged_group,unprivileged_groups)*	Set groups	Evaluation	Group design
AIF.360	*aif360.sklearn.metrics.intersection()*	Intersectional groups	Evaluation	Group design
FairLearn	*MetricFrame.difference()*	Max. absolute difference	Evaluation	Meta-measure
Aequitas	*Fairness.get_group_attribute_fairness()*	Get unbiased decision	Evaluation	Meta-measure

Table 4.2: Examples of pattern instantiations in software tools

to achieve a similar objective. The tool outputs a table of binary True/False values for each group attribute (e.g. age, gender, race). The output value is False (i.e. the model is biased) if the bias measure of at least one group attribute value (e.g. male, female) indicates disparate performance. While these instantiations differ, they can both be consolidated in the *Meta-measure* pattern, which is concerned with aggregating bias measures across groups into a single value to determine if a model is biased.

4.4. Pattern Catalogue

This section introduces the patterns for detecting and mitigating bias in ML. The section first presents the 23 unique patterns that were consolidated from 106 pattern instances, identified through bottom-up analysis of software tools. Following the GOF convention, this collection of all patterns is called a catalogue [93]. Next, the conceptual model is used to organise the patterns by their purpose and scope. Finally, the section presents a statistical analysis of pattern occurrences in the analysed software tools.

4.4.1. The Patterns

The patterns represent the lowest-level actions for bias detection and mitigation. They have been formulated as *name/key idea* pairs and are listed in alphabetical order by their name. Each key idea is expressed as a succinct statement that conveys useful information that is a building block for detecting or mitigating bias. As discussed in the previous section, this form offers greater flexibility than fixed format templates. An extend form may emerge once the patterns are validated in use cases.

Adversarial bias mitigation
Learn a classifier or regressor to maximise predictive performance and simultaneously reduce an adversary's ability to determine a protected or sensitive attribute from the predictions.

Continuous testing
Integrate bias tests into the development process and continuous testing to ensure that ML systems make equitable predictions after they have been launched.

Data quantity distribution across groups
Ensure that all groups are represented sufficiently in the data.

Decorrelating sensitive features
Apply a linear transformation to non-sensitive features to remove their correlation with the sensitive feature while retaining as much information as possible.

Disaggregated base metrics
Calculate performance metrics across groups.

Error Tolerance
Specify a tolerable margin for error rates between groups and base the decision on whether a model is biased or not on this threshold.

Evaluation datasets
Evaluate models on multiple, representative datasets that capture the application context.

Feature editing
Apply an algorithm to edit feature values across groups to reduce bias measures subject to constraints and optimisation objectives.

Fairness assumptions
Make assumptions explicit with an upfront statement about that which is believed to make a system fair.

Ground-truth label distribution across groups
Ensure that ground-truth labels are balanced across groups.

Group design
Capture the rationale for choosing attributes that define groups and for deciding when enough different groups have been evaluated.

Group-based bias measure
Compare base metrics across groups or between groups and overall performance to assess whether different groups are treated equally.

Imposing fairness constraints on the optimisation objective
Add a bias measure to the regularisation term of the learning objective.

Individual bias measure
Compare base metrics across individuals to assess whether similar individuals receive similar predictions.

In-processing model selection
Incorporate a bias constraint during hyperparameter optimisation to search for and select models that minimise the error rate and bias constraint.

Metadata approximation
Approximate metadata labels from other features in the dataset.

Meta-measure
Aggregate bias measures across groups into a single bias measure for the model to compare it against other models.

Output label redistribution
Change output labels to meet a bias-reduction objective.

Sampling and reweighing
Balance the training data across groups and labels through sampling or by applying weights.

Threshold evaluation
Evaluate metrics across multiple thresholds in order to understand how the threshold can affect the performance for different groups.

Threshold selection
Adjust the model's output by selecting thresholds that take into account a predictive performance objective subject to constraints that limit bias.

Types of harm
Identify context and application specific risks and consequently harm that a biased model can lead to.

Visualising Metrics and Measures
Plot disaggregated base metrics and bias measures to visually examine performance disparities.

4.4.2. Organising the Pattern Catalogue

The patterns have been categorised along two axes that designate their overarching purpose and scope in Table 4.3. This categorisation corresponds with the conceptual model presented in Figure 4.2. The pattern purpose is either analysis or design, depending on whether the pattern is used for bias detection or mitigation. The pattern scope is determined based on the stage of the ML workflow in which the pattern would typically be used. Patterns that fall within an evaluation scope and that are used for the purpose of analysis represent the largest category of patterns found in toolkits.

Scope	Purpose	
	Analysis	**Design**
Dataset	Data quantity distribution across groups Ground-truth label distribution across groups Metadata approximation	
Evaluation	Continuous testing Disaggregated base metrics Error tolerance Evaluation datasets Fairness assumptions Group design Group-based bias measure Individual bias-measure Meta-measure Types of harm	
Pre-processing		Decorrelating sensitive features Feature editing Sampling and reweighing
In-processing		Adversarial bias mitigation Imposing fairness constraints on the optimisation objective In-processing model selection
Post-processing		Output label redistribution Threshold evaluation Threshold selection
Interaction	Visualising Metrics and Measures	

Table 4.3: Categorisation of patterns based on purpose and scope

4.4.3. Analysis of Patterns in Software Tools

Prior research pointed out that the functionalities and limitations of toolkits for bias detection and mitigation oftentimes constrain the scope of a bias analysis, as bias evaluations are frequently reformulated to fit a format for which current toolkits provide support [56]. Gaining insights into the functionalities that are supported by toolkits is thus important to understand how tool capabilities may restrict or shape bias evaluations. However, eliciting tool capabilities is challenging. Patterns offer a useful framing for analysing and comparing the capabilities of software tools, and can shed light on areas where tooling is lacking in practice. To provide insights on how toolkits may constrain bias evaluations, this section analyses the frequency of pattern occurrences in the selected software tools to establish which patterns are frequently implemented and to identify patterns and pattern categories that are not well supported. First, the frequency of occurrence of pattern scope, and then the frequency of occurrence of patterns is analysed across toolkits.

Frequency of Occurrence of Pattern Scopes

Figure 4.5 visualises how frequently patterns of each scope occur across software tools. Across software tools evaluation patterns occur most frequently. This is not surprising, as evaluation patterns are required for detecting bias, and bias detection is necessary for bias mitigation as illustrated in Figure 4.3. Evaluation patterns are thus broadly applicable. In addition to their central role in bias detection and mitigation, evaluation patterns can also be instantiated in a wide variety of ways. For example, the Fairness Indicators tool has over 50 metrics to evaluate many different tasks and problems. Each of these metrics, when calculated across groups, is an instantiation of the *Disaggregated base metrics* pattern. Similarly, the AI Fairness 360 tool can calculate ratios or differences across a variety of base metrics, thus supporting a variety of instantiations of the *Group-based bias measure* pattern.

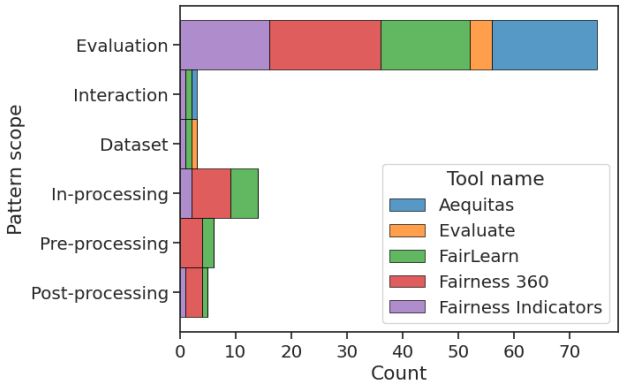


Figure 4.5: Frequency with which patterns of a given scope have been instantiated in toolkits.

In-processing patterns occur second most frequently. This may be because in-processing bias mitigation has received more attention in the academic literature than pre- and post-processing approaches [230]. It is surprising that dataset patterns do not occur more frequently. Like evaluation patterns, they play a central role in bias detection and mitigation. A possible reason for this could be that some dataset patterns, like *Data quantity distribution across groups* and *Ground-truth label distribution across groups*, can be instantiated with methods available in generic software libraries that are not specific to bias detection and mitigation. However, even if this is possible, it does not preclude the utility of dataset patterns. Lee and Singh [166] raised the absence of dataset related toolkit features (e.g. the ability to check whether a dataset is representative of the broader population, or whether a dataset contains proxy-variables) as a gap in existing software tools. Given the limitations of the toolkits, it is likely that the dataset patterns identified in Section 4.3 are incomplete and that new dataset patterns will be added in future.

Frequency of Occurrence of Patterns

Figure 4.6 shows the occurence of the 10 most frequent patterns. The top four patterns are evaluation patterns, with the *Disaggregated base metric* and *Group-based bias measure* patterns occurring most often. Only the *Disaggregated base metric* pattern occurs in all tools. With the exception of the Evaluate tool, all software tools include the *Group design* pattern. The *Visualising metrics and measures* pattern (interaction), *Imposing fairness constraints on the optimisation objective pattern* (in-processing) and *Types of harm* pattern (evaluation) occur in three out of the five software tools. Four patterns are implemented in only two or even a single software tool. This analysis confirms observations in the literature that the bias detection and mitigation functionalities of tools differ substantially [166]. Using the patterns to compare the tools gives clearer insights into how they differ, and which gaps in tooling may impact bias evaluations.

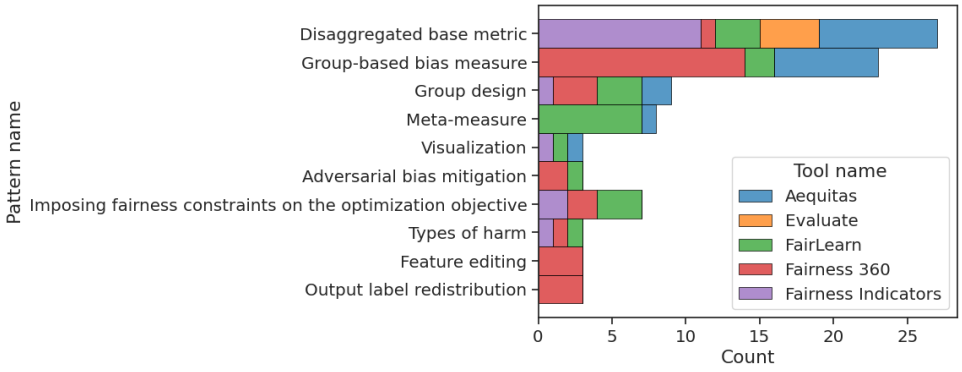


Figure 4.6: Frequency with which the top 10 patterns have been instantiated in software tools.

4.5. Capturing Processes as Pattern Recipes

Measuring bias and curating benchmark datasets are two essential processes for conducting bias evaluations and taking intervening actions. This section introduces the *Bias Measurement Recipe* and the *Benchmark Dataset Recipe* for guiding these processes. As illustrated in Figure 4.1, the recipes are created from collections of patterns and capture tacit knowledge I gained through personal practice. In the chapters that follow, the recipes are used and evolved through application in projects.

4.5.1. Recipes from Pattern Collections

Individually, each pattern contributes an idea towards detecting or mitigating bias but it cannot offer a complete bias evaluation or intervention on its own. To achieve this end goal, patterns need to be assembled into collections and used together. A collection of patterns can guide practitioners through the steps that are necessary to detect or mitigate bias reliably and consistently.

Recipes Guide Processes

The purpose of using patterns in a collection is similar to using a recipe when cooking. A recipe specifies how a meal should be made by listing ingredients and the steps to follow in preparing and combining them. Recipes vary just as much as the meals that can be made. They use different ingredients, some essential, others optional. The way ingredients are prepared can change. At times the order in which ingredients are added is critical, at other times it does not matter. Following a recipe reduces the risk of making a meal that is distasteful, and maximises the likelihood of making a delicious meal. New recipes can be invented, but must withstand the test of taste.

Pattern Recipes and Notation

A recipe of patterns consists of a set of patterns that, when used together, achieve a common bias detection or mitigation objective. As with cooking, pattern recipes guide processes. A pattern recipe consists of constituent patterns, and guidelines on how they should be used together. Patterns can be required or optional members of a recipe, and may need to be applied in a specific order. By following a recipe, a practitioner ensures that their process for bias detection or mitigation is repeatable, coherent, consistent and comprehensive. Like patterns, recipes should be named so that they can become part of the vocabulary of stakeholders that partake in bias detection and mitigation. In addition to the name, a recipe also requires a list of patterns. The convention introduced in this thesis for listing patterns is to first enumerate and list patterns where the order of application matters, and then list unordered patterns. Furthermore, required patterns are marked with '+' and optional patterns with 'o'. For optional patterns the ordering only applies if they are used. Unordered patterns will depend on one or more ordered patterns and are typically used after the ordered patterns.

Proactive and Retrospective Modes of Using Pattern Recipes

Recipes can be used in different ways. If used proactively to achieve a particular purpose, such as measuring bias or creating a benchmark dataset, a recipe functions as a design tool. In this mode the key ideas of the patterns, formulated as statements, can be interpreted as design instructions – "*take this action to make progress towards detecting/mitigating bias*". For this reason this thesis calls this the *proactive* mode of a recipe. Alternatively, a recipe can be applied retrospectively to interrogate and evaluate the quality of an existing process, for example a bias evaluation or benchmark dataset. In this mode the recipe functions as an analysis tool, and is operating in *retrospective* mode. In retrospective mode the patterns can be used more intuitively if they are reframed as questions – "*has this action been taken?*" – or as an analysis instruction. Reframing the patterns for the retrospective mode should help answer the following questions:

1. Has this pattern been applied intentionally and sufficiently?
2. Does this pattern point to a source of bias?

4.5.2. Bias Measurement Recipe

Bias Measurement

- o 1 Types of harm
- o 2 Fairness assumptions
- + 3 Group design
- o 4 Metadata approximation
- + 5 Disaggregated base metrics
- o 6 Group-based bias measure
  - o Visualising metrics and measures
  - o Meta-measure

The *Bias Measurement* recipe is a collection of patterns that are necessary or useful when measuring bias. The recipe is important, as bias measurement determines how performance disparities in ML systems are quantified. The recipe consists of two required and six optional patterns. The two required patterns are *Group design*, followed by *Disaggregated base metrics*, as bias measurement cannot be done without these two patterns. The remaining patterns are optional. The first six patterns are ordered. The *Visualising metrics and measures* and *Meta-measure* are unordered and can be used after the ordered patterns. While *Types of harm* and *Fairness assumptions* are optional, if used they should come first, as they frame the overall evaluation. They are optional patterns, as bias measurement can be done even if harms are not considered explicitly and fairness assumptions are not stated. However, a bias measurement that does not consider harms or state its assumptions is evidently more limited in its ability to draw conclusions about societal consequences and fairness than one that includes those patterns.

4.5.3. Benchmark Dataset Recipe

Benchmark Dataset
+ 1 Evaluation datasets
+ 2 Group design
o 3 Metadata approximation
  + Ground-truth label distribution across groups
  + Data quantity distribution across groups

The *Benchmark Dataset* recipe is a collection of patterns that ensure that benchmark datasets used for evaluation are representative, appropriate and sufficient. The recipe is important, as benchmarks determine the scope and context of a bias evaluation, and thus its limits in relation to a specific application. Moreover, benchmarks often have the propensity to amplify bias, as they shape evaluation habits and implicitly guide the culture of technology development in a domain [127, 226]. The recipe consists of four required and one optional pattern. The *Metadata approximation* pattern remains optional, as it may not always be possible or desirable to approximate metadata. The first three patterns are ordered. The *Ground-truth label distribution across groups* and *Data quantity distribution across groups*, while required, can be applied in either order after the enumerated patterns.

4.6. Discussion of Pattern Validity and Limitations

This section considers the validity of the identified patterns and recipes in relation to the requirements set out in Section 4.2.4. All patterns are considered to meet Requirements 1 – 4, 10 and 12 – 17. The requirements that concern pattern function will be assessed in use cases in Chapters 5 and 6. Requirement 5 needs to be validated in future work. The section furthermore discusses the limitations of the approach applied to identifying patterns in this research.

Requirement(s)	Type	Validated in
1 - 4	**Form**	**this chapter**
5	Form	not validated in this thesis
6 - 8	Function	Chapter 5
9	Function	Chapter 6
10, 12 - 13	**Content**	**this chapter**
11	Content	Chapter 6
14 - 17	**Interaction**	**this chapter**

Table 4.4: Overview of requirements validation

4.6.1. Validating Patterns Against Requirements

Requirements 1 – 4 are concerned with pattern form. Meeting these requirements was a key consideration while formulating patterns as *name/key idea* pairs. All patterns

have thus been formulated to be unique, have a name and express a single idea that supports bias detection or mitigation. Great care was taken to meet Requirement 4 by expressing patterns as simply as possible without oversimplifying them or loosing contextual nuance. However, many ways exist to formulate patterns in such a manner, and the validation of Requirement 4 is thus subjective.

Identifying patterns from software tools was an explicit choice that was made to ensure that patterns have been found useful or necessary for detecting or mitigating bias in one or more practical contexts. The underlying assumption is that tool creators would not have built tools that have no utility to practitioners, and that patterns derived from programmed methods and documented concepts in tools and documentation are thus useful or necessary. While the instantiation of a pattern in a tool does not prove its utility in practice, it does validate that the pattern was deemed to be useful or necessary by tool creators. In this early stage of pattern development, this is sufficient to consider all patterns as meeting Requirement 10. By using the conceptual model to guide pattern consolidation and categorise patterns by scope based on the ML development workflow, all patterns are applicable to a stage of the ML development workflow and meet Requirement 12.

Requirements 13 – 16 were accounted for while consolidating patterns from pattern instances found in software tools. During this process it was ensured that patterns are not instantiations of a particular method or specific to a programming language. Each pattern was validated as being meaningful on its own or when used with other patterns. The potential of a pattern to detect and mitigate bias was subjectively evaluated, guided by personal experience. If a pattern was deemed to be unable to detect and mitigate bias on its own, its potential contribution to a collection of patterns was considered instead. For example, the *Group design* pattern is insufficient for detecting or mitigating bias on its own, but it is an important contributor to the *Bias Measurement* and *Benchmark Dataset* recipes and is thus a valid pattern. Patterns were also checked to be free from contradictions and consistent with each other.

The two pattern recipes that were created serve the particular objectives of measuring bias and ensuring that benchmark datasets used for evaluation are representative, appropriate and sufficient. They thus achieve a particular bias detection or mitigation objective and meet Requirement 17.

Patterns that have not been validated in this chapter

Validating Requirement 5 requires a user study with practitioners that have limited experience in bias detection and mitigation. This kind of validation is not conducted in this thesis and remains an area of future work. Requirements 6, 7, 8 and 9 concern pattern function and will be validated together with Requirement 11 in use cases in the next chapters. The use case in Chapter 5 is focused on the validation of pattern function as set out in Requirements 6, 7 and 8, while Chapter 6 validates the reusability and adaptability of patterns to a different application context, as called for by Requirements 9 and 11. As the patterns identified in this chapter are not considered to be complete,

further patterns can be identified in the use cases. Like the patterns identified in this chapter, they will need to be validated against the requirements.

4.6.2. Limitations of the Approach

Choosing to identify patterns from programmed methods in software tools and their accompanying user documentation constrained the patterns that could be identified in this research. The analysis of pattern instantiations across software tools in Section 4.4.3 shows that not all pattern categories are well represented in current toolkits. The representation of patterns in the catalogue is skewed as a result. Patterns with an analysis purpose that are necessary for evaluation have been instantiated most frequently in current tools. Particularly the *Disaggregated base metric*, *Group-based bias measure* and *Group design* patterns occur in all or most tools. Patterns with a dataset or interaction scope, on the other hand, are not well represented. As prior literature pointed out, practitioners oftentimes reformulate their bias evaluations to fit toolkit capabilities [56]. The representation bias in the identified patterns thus also points to areas where further tooling is required, and highlights opportunities for future research.

The conceptual model that was used to guide the categorisation and consolidation of pattern instances into patterns facilitated a methodological process that made pattern identification tractable. However, it also limited the patterns that were identified to those that align with the scope categories. Moreover, as tacit personal experience was used to guide how patterns were consolidated and formulated from the programmed methods and documentation, this process remains subjective and one that will be difficult to reproduce and repeat exactly. The study also limited pattern identification to 5 software tools, which may introduce selection bias. While more of the tools listed in Table 4.1 can be analysed to identify further patterns, it was considered important to first validate if the existing patterns meet functional requirements before identifying more patterns. In addition to analysing more tools, future work should also consider other ways of eliciting patterns, for example by interviewing practitioners and tool creators, and by doing a systematic review of the literature on the topic.

4.7. Conclusion

This chapter launched the design science research cycle, and set out to answer RQ1, namely to determine which established approaches for bias detection and mitigation in ML fairness can help researchers and practitioners from other domains detect and mitigate bias in ML. The key contributions of the chapter are a conceptual model for categorising patterns, a catalogue of 23 patterns for detecting and mitigating bias in ML, and two pattern recipes for measuring bias and curating benchmark datasets.

The chapter defines a pattern for detecting or mitigating bias in ML as "*an idea*

that has been found useful or necessary for detecting or mitigating bias in one practical context, and is likely to continue being useful or necessary for that purpose in other contexts." The patterns are formulated as *name/key idea* pairs, and were derived through a bottom-up analysis of software tools that detect and mitigate bias in ML. The catalogue organises the patterns along two dimensions: their purpose and scope. These dimensions were introduced in the conceptual model, which was proposed to guide the pattern discovery process. The pattern purpose can be either one of analysis or design, depending on whether a pattern serves the purpose of bias detection or mitigation. The pattern scope approximates the stages of the ML workflow: pre-processing, in-processing, post-processing, dataset, evaluation and interaction.

The recipes capture tacit knowledge from personal practice and demonstrate how low-level patterns can be used together to guide important processes for bias detection and mitigation. Naturally, recipes thus require a list of constituent patterns and like the patterns, they are named. The chapter introduced two recipes that are important for bias detection and mitigation. The *Bias Measurement* and *Benchmark Dataset* recipes guide how performance disparities in ML systems are quantified, and ensure that benchmark datasets used for evaluation are representative, appropriate and sufficient. The chapter introduced a convention for listing patterns in a recipe that takes the order of use and whether patterns are required or optional into account. These recipes thus make the relationships and dependencies between patterns explicit, and allow practitioners to identify and account for them when they design processes for detecting and mitigating bias.

This chapter successfully identified and captured relevant knowledge and practices from ML fairness as design patterns for detecting and mitigating bias in ML. It thus addresses RQ1 and accomplishes Goal 1. In the next chapter the functional requirements of patterns and recipes will be validated in a use case to examine how collections of patterns can support practitioners in detecting and mitigating bias in ML.

5

Validating Pattern Utility

This chapter validates the functional requirements of the identified patterns in a machine learning (ML) use case. Focusing on speaker verification, the use case is based on research that presented the first comprehensive study on bias detection and mitigation in the ML development workflow in this domain. The chapter makes two main contributions to the thesis. Firstly, the Benchmark Dataset and Bias Measurement recipes and their constituent patterns are shown to enable coherent, reproducible and consistent bias detection that supports the interrogation of the impacts of design choices in a speaker verification benchmark. Secondly, the chapter identifies approaches to mitigate pitfalls in speaker verification evaluations by applying the patterns and recipes. The insights gained through this research informed two design iterations that introduced two new patterns, and through which more detailed templates emerged to guide practitioners in using the patterns.

This chapter draws on the following publications:

1. W. Hutiri and A. Y. Ding. Bias in Automated Speaker Recognition. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, pages 230–247, Seoul, Republic of Korea, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533089 [128]
2. W. Hutiri, L. Gorce, and A. Y. Ding. Design Guidelines for Inclusive Speaker Verification Evaluation Datasets. In *Interspeech 2022*, Incheon, Republic of Korea, 2022. International Speech Communication Association. doi: 10.21437/Interspeech.2022-10799 [130]

5.1. Introduction

This chapter validates the patterns and recipes identified in Chapter 4 against functional requirements 6, 7 and 8. This is done by applying the *Benchmark Dataset* and *Bias Measurement* recipes, which use patterns to capture processes for bias detection and mitigation, in a use case in the speaker verification domain. At the time this research started, no studies on bias had been conducted in speaker verification. The chapter thus demonstrates the utility of the patterns and recipes in a domain where limited prior research has studied approaches to detecting and mitigating bias.

Speaker recognition is a type of voice processing that automatically recognises the identity of a human speaker from personal information contained in their voice [91]. Core tasks in speaker recognition are speaker identification, which determines the identity of a speaker from a subset of speakers, speaker verification, which validates if a speaker's identity matches the identity of a stored speech utterance, and speaker diarisation, which is concerned with partitioning speech to distinguish between different speakers [11]. This chapter focuses on automatic speaker verification. Automatic speaker verification technology underlies biometric voice identification systems, which offer a hidden and passive way to authenticate people. In voice-activated Edge AI, speaker verification thus plays an important role to authenticate users and secure a system from intruders [129].

Bias and discrimination in face recognition [30, 238, 239], natural language processing [25] and automatic speech recognition [2, 155, 282, 292] are well studied and documented. Bias in speaker verification, on the contrary, has received very limited attention in the literature. Yet, speaker verification systems are sensitive to a broad range of socio-demographic factors [87, 108, 335], for example age, spoken language, speaking style and rate, gender, accent, emotion and health conditions. These factors can lead to performance disparities in speaker verification systems, which then result in disparate voice biometrics service quality for technology users. Consequently, it is important to detect bias in speaker verification, so that mitigating actions can be taken and harmful effects of service failure can be forestalled.

This chapter uses recipes to detect and mitigate bias in a popular speaker verification benchmark, the VoxCeleb Speaker Recognition Challenge. The chapter starts with an overview of speaker verification in Section 5.2. In Section 5.3 the pattern recipes are used to detect bias and a new dataset pattern, *Data source*, is proposed. Section 5.4 examines pitfalls in speaker verification evaluation datasets and proposes the new *Data quality distribution across groups* dataset pattern to mitigate these shortcomings and improve the *Benchmark Dataset* recipe. Next, Section 5.5 investigates pitfalls in speaker verification bias measurements, showing how unexpected contradictions can arise when bias evaluations do not pay careful attention to the *Bias Measurement* recipe. The chapter concludes with a summary of key contributions in Section 5.6.

5.2. Speaker Verification Use Case

This section provides a high level overview of speaker verification and its evaluation, as well as its supporting ecosystem of competitions, challenges and benchmarks that have advanced the field. For a detailed technical survey on state-of-the-art speaker recognition refer to [11], and to [151] for a review on the classical speaker recognition literature prior to the advent of Deep Neural Networks (DNNs).

5.2.1. Speaker Verification Overview

A speaker verification system determines whether a candidate speaker matches the identity of a registered speaker by comparing a candidate speaker's speech signal (i.e. *trial utterance*) to the speech signal of a registered speaker (i.e. *enrollment utterance*). Speaker verification is classified based on its training data as text-dependent if speech signals are fixed phrases (e.g. "my voice is my password") or text-independent if not, prompted if speech was produced by reading text or spontaneous if not [101]. Spontaneous text-independent speech is the type of speech that occurs naturally when a person interacts with a voice assistant or a call centre agent, and presents the most general speaker verification task setting.

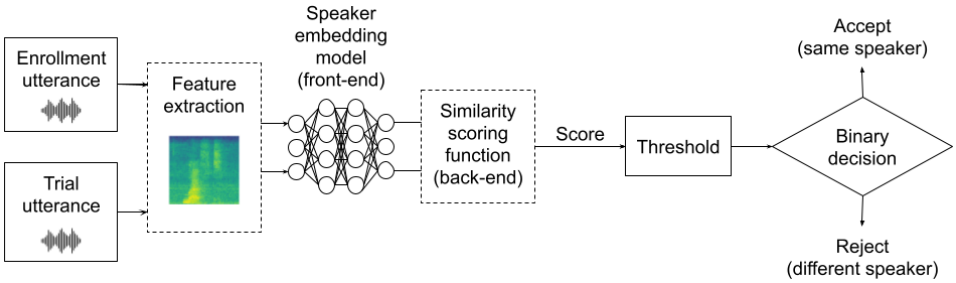


Figure 5.1: Speaker verification data processing pipeline

As shown in Figure 5.1, many speaker verification systems consist of two stages; a front-end that generates a speaker embedding model for enrollment and trial utterances, and a back-end that computes a similarity score for the two resultant embeddings. Alternatively, end-to-end speaker verification directly learns a similarity score from training utterances [114]. Modern speaker verification systems use DNNs to learn the front-end embedding, or to train the end-to-end system [11]. As the final step of the speaker verification process, the score output is compared to a threshold. Speaker identity is accepted if the score lies above the threshold, and rejected if it lies below the threshold.

5.2.2. Speaker Verification Evaluation

To evaluate speaker verification systems, scores are generated for many pairs of enrollment and trial utterances. The utterance pairs are labelled as being from the same or from different speakers. Two typical score distributions generated from many same and different speaker utterance pairs are shown in Figure 5.2. After calibrating the speaker verification system to a threshold, utterance pairs with a score below the threshold are classified as different speakers and the trial utterance is rejected. Utterance pairs with a score above the threshold are classified as the same speaker, and accepted. As the two distributions overlap, classification is not perfect. At a particular threshold value there will be false positives, i.e. utterance pairs of different speakers with a score above the threshold, and false negatives, i.e. utterance pairs of the same speakers with a score below the threshold.

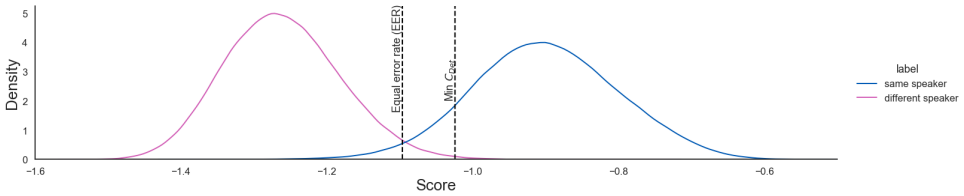


Figure 5.2: Example distribution of speaker verification scores: blue are same speaker trials, pink are different speaker trials. The dotted lines are possible threshold values. Scores to the left of a threshold are rejected, scores to the right are accepted.

The performance of a speaker verification system is determined by its false positive rate (FPR) and false negative rate (FNR) at the threshold value to which the system has been calibrated [101]. It is accepted that the two error rates present a trade-off, and that selecting an appropriate threshold is an application-specific design decision [217]. The threshold value is determined by balancing the FPR and FNR error rates for a particular objective. Popular objectives are obtaining an *equal error rate* (EER) for FPR and FNR, or minimising the detection cost function (DCF), a weighted sum of FPR and FNR errors calculated at different threshold values. The detection cost, $C_{Det}(\theta)$, is the value of the DCF at a particular threshold θ. Its minimum value, $min\ C_{Det}$, is often reported as a metric. Various DCF have been proposed over time. This chapter uses the DCF proposed in the NIST SRE 2019 Evaluation Plan [216]:

$$C_{Det}(\theta) = C_{FN} \times P_{Target} \times P_{FN}(\theta) + C_{FP} \times (1 - P_{Target}) \times P_{FP}(\theta)$$
$$P_{Target} = 0.05,\ C_{FN} = 1,\ C_{FP} = 1 \tag{5.1}$$

Prior research in speaker recognition recommends that system performance is considered across various thresholds and that trade-offs between FPR and FNR are considered [101]. *Detection error trade-off* (DET) curves visualise the FPR and FNR at different operating thresholds on the x- and y-axis of a normal deviate scale [183] (see Fig-

ure 5.3). DETs can be used to analyse inter-model performance across models, and are also recommended for analysing intra-model performance across speaker subgroups in a model.

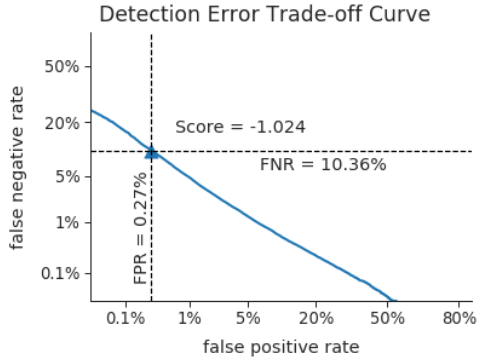


Figure 5.3: Example detection Error Trade-off (DET) curve of a speaker verification system: the blue line shows false positive and false negative error rates at different score values. For example, at the blue triangle the score = -1.024, FPR = 0.27% and FNR = 10.36%

5.2.3. Speaker Verification Benchmarks

Speaker recognition challenges have played an important role in evaluating and benchmarking advances in speaker verification. They were first initiated within the Information Technology Laboratory of the US National Institute of Standards and Technology (NIST) to conduct evaluation driven research on automated speaker recognition [101]. The NIST Speaker Recognition Evaluation (SRE) challenges and their associated evaluation plans have been important drivers of speaker verification evaluation. In addition, new challenges have emerged over time to address the requirements of new applications and tasks.

Name	Organiser	Years	Metrics
NIST SRE [101]	US National Institute of Standards & Technology	1996 - 2021	Detection Cost Function
SRE in Mobile Env's [146]	Idiap Research Institute	2013	DET curve, EER, *half total error rate*
Speakers in the Wild SRC [191]	at Interspeech 2016	2016	$min\ C_{Det}$* (SRE2016), R_{prec}, C_{llr}
VoxCeleb SRC [208]	Oxford Visual Geometry Group	2019 - 2021	$min\ C_{Det}$* (SRE2018), EER
Far-Field SVC [236]	at Interspeech 2020	2020	$min\ C_{Det}$*, EER
Short Duration SVC [331]	at Interspeech 2021	2020 - 2021	$norm\ min\ C_{Det}$* (SRE08)
SUPERB benchmark [328]	CMU, JHU, MIT, NTU, Facebook AI	2021	EER*

Table 5.1: Evaluation metrics for Speaker Verification and Recognition Challenges (SVC and SRC) (* primary metric)

Table 5.1 summarises recent challenges, their organisers and the metrics used for evaluation. Many competitions, such as the SITW [191], the VoxCeleb SRC [208] and the Far-Field SRC [236] have been hosted at research conferences in the speech domain. The competitions provide benchmarks, evaluation protocols and baseline models that researchers and practitioners use to develop and test their systems. Most challenges have adopted the minimum detection cost ($min\ C_{Det}$) or the equal error rate (EER) as their primary performance metric. The VoxCeleb SRC stands out as a challenge that has gained a lot of traction in recent years, and its benchmark datasets now dominate evaluations in speaker recognition research [249].

5.3. Using Pattern Recipes to Detect Bias

This section uses the *Benchmark Dataset* and *Bias Measurement* recipes in an analytical and empirical study to detect bias in the VoxCeleb SRC baseline models and the VoxCeleb 1 benchmark dataset. The section draws on my prior work, published in [128]. The section starts with details of the study setup, then uses the *Benchmark Dataset* recipe and after that the *Bias Measurement* recipe to identify multiple sources of bias embedded in the VoxCeleb SRC. The *Benchmark Dataset* recipe is used in retrospective mode to detect sources of bias in evaluation sets derived from VoxCeleb 1, which are frequently used to benchmark speaker verification models. The *Bias Measurement* recipe is used in proactive mode to conduct a bias evaluation of baseline models released by the VoxCeleb SRC.

5.3.1. Study Setup

Empirical experiments were conducted on two pre-trained models, which were used as black-box predictors. The code for the study has been released online[1].

VoxCeleb 1 Evaluation Sets

VoxCeleb 1 [209] was released in 2017 to explore the use of deep neural networks (DNNs) for speaker recognition tasks [207]. The goal of the dataset creators was thus to gather a text-independent, large scale speaker recognition dataset that mimics unconstrained, real-world speech conditions. The dataset contains 153 516 short clips of audio-visual utterances of 1251 celebrities in challenging acoustic environments (e.g. recordings with background chatter, laughter and speech overlap) extracted from YouTube videos. The dataset also includes metadata for speakers' gender and nationality. The VoxCeleb SRC recommends three evaluation sets for benchmarking speaker verification models: VoxCeleb 1-test, VoxCeleb 1-E and VoxCeleb 1-H. These sets are constructed by pairing utterances of individual speakers in VoxCeleb 1 with themselves and with different speakers based on matching criteria.

[1] https://github.com/wiebket/bt4vt/releases/tag/v0.1

Baseline Model Architectures and Training Dataset

The VoxCeleb SRC has released two baseline models [116] which were pre-trained on the VoxCeleb 2 training set [209] with close to 1 million speech utterances of 5994 speakers. 61% of speakers are male and 29% of speakers have a US nationality, which is the most represented nationality. More detailed metadata for the training set is not readily available. VoxCeleb 2 is disjoint from VoxCeleb 1. The baseline models are based on a 34-layer ResNet trunk architecture. Further technical details are summarised in Table 5.2. *ResNetSE34V2* is a larger model, with an architecture optimised for predictive performance. *ResNetSE34L* is a smaller model optimised for fast execution. It contains less than a fifth of the parameters of *ResNetSE34V2* and has smaller input dimensions. This reduces the computation time and memory footprint of the model, which are important considerations for on-device deployment in applications like smartphones and smart speakers.

Model	***ResNetSE34V2***	***ResNetSE34L***
Published in	Heo et al. [116]	Chung et al. [43]
Alternative name in publication	performance optimised model, H/ASP	Fast ResNet-34
Additional training procedures	data augmentation (noise and room impulse response)	-
Parameters	8 million	1.4 million
Frame-level aggregation	attentive statistical pooling	self-attentive pooling
Loss function	angular portotypical softmax loss	angular portotypical loss
Input features	64 dim log Mel filterbanks	40 dim Mel filterbanks
Window (width x step)	25ms x 10ms	25ms x 10ms
Optimized for	predictive performance	fast execution

Table 5.2: Technical attributes of the VoxCeleb SRC baseline models

5.3.2. Detecting Bias with the Benchmark Dataset Recipe

The *Benchmark Dataset* recipe contains patterns that aide in interrogating sources of bias in benchmark and evaluation datasets. The recipe as introduced in Chapter 4 includes five patterns. This analysis introduces an additional pattern, *Data Source*, as the data curation process of the dataset is a significant contributor to bias. The recipe is used in retrospective mode to analyse sources of bias in the VoxCeleb 1 evaluation sets, and exposes their limits for assessing the performance

Benchmark Dataset Recipe

+ 1 Evaluation datasets
+ 2 Group design
o 3 Metadata approximation
  + Ground-truth label distribution across groups
  + Data quantity distribution across groups
  + *Data source (new)*

of speaker verification systems. Next, I discuss how each pattern has been applied to show how it supports bias detection and the interrogation of design choices.

Evaluation datasets

Evaluate models on multiple, representative datasets that capture the application context.

The pattern is reframed as a question: *Are the VoxCeleb 1 evaluation sets representative and capture relevant application contexts?* The pattern analysis includes the following sections:

1. Attributes of evaluation sets
2. Representativeness of evaluation sets
3. Context captured by evaluation sets
4. Consequences for evaluation
5. Critical reflection on evaluation datasets

---

1. Attributes of evaluation sets

The attributes, speaker and utterance level statistics for each evaluation set are summarised in Table 5.3. *VoxCeleb 1-test* contains utterance pairs of 40 speakers whose name starts with *E*. *VoxCeleb 1-E* includes the *entire* dataset, with utterance pairs sampled randomly. *VoxCeleb 1-H* is considered a *hard* test set, that contains only utterance pairs where different speakers have the same gender and nationality. Speakers have only been included in *VoxCeleb 1-H* if there are at least 5 unique speakers with the same gender and nationality.

Attribute	VoxCeleb 1-test	VoxCeleb 1-E	VoxCeleb 1-H
unique speakers	40	1 251	1 190
unique utterance pairs	37 720	579 818	550 894
speaker pairing details	-	random sample	same gender, nationality
speaker pair inclusion criteria	name starts with 'E'	all	>=5 same gender, nationality speakers
female / male speakers (%)	38 / 62	45 / 55	44 / 56
female / male utterances (%)	29.5 / 70.5	41.8 / 58.2	41.1 / 58.9
count of nationalities	9	36	11
top 1 nationality (% spkrs / utt.)	US (62.5 / 59.6)	US (63.9 / 61.4)	US (67.1 / 64.7)
top 2 nationality (% spkrs / utt.)	UK (12.5 / 13.9)	UK (17.2 / 18.3)	UK (18.1 / 19.3)
top 3 nationality (% sprks / utt.)	Ireland (7.5 / 6.7)	Canada (4.3 / 3.8)	Canada (4.5 / 3.9)

Table 5.3: VoxCeleb 1 evaluation sets show that the benchmark's population is not representative across gender and nationality

2. Representativeness of evaluation sets

Several observations can be made from Table 5.3. All three evaluation sets overrepresent male speakers and US nationals. The inclusion criterion of *VoxCeleb 1-test*, which

only includes speakers whose name starts with an 'E', is arbitrary and a source of bias as names strongly correlate with language, culture, gender and ethnicity. Moreover, the sample size of *VoxCeleb 1-test* is very small, making its validity for evaluation questionable. While *VoxCeleb 1-E* is considerably larger, its random sample pairing strategy cannot control the difficulty of different speaker pairings in the evaluation set. This can result in some pairs being significantly easier to compare than others, and lead to unequal evaluations across speaker groups. *VoxCeleb 1-H* controls the difficulty of different speaker pairings, but does not control the representativeness of the evaluation set within and across speaker groups.

Nationality and gender only account for some of the attributes of the human voice that affect speaker verification [271]. Further analysis of the evaluation set representativeness is limited by available metadata labels. However, a recent metadata extensions of VoxCeleb 1 by speaker's age shows that people between ages 20 and 50 are overrepresented in the dataset [113].

3. Context captured by evaluation sets

Evaluation sets should contain diverse types of people and speaking conditions that resemble those of the application context. Sourced from celebrity recordings, VoxCeleb 1 does not capture the people that typically use speaker verification technology, or the contexts in which it is used. For example, voice-activated Edge AI as found in voice assistants in homes, cars, offices and public spaces, is highly localised. People using these systems will frequently share a nationality, language, accent and possibly even DNA. Their voices will thus be much more similar than those of randomly paired speakers. These user and usage contexts should be reflected in speaker verification evaluation sets. *VoxCeleb 1-test* and *-E*, which have randomly paired utterances, are inadequate to capture speaker verification performance in voice-activated Edge AI application scenarios. Even *VoxCeleb 1-H* cannot evaluate typical voice assistant scenarios, such as the ability of a speaker verification system to distinguish family members.

4. Consequences for evaluation

Empirical results in Figure 5.4 illustrate that the evaluation sets impact the evaluation outcomes. The figure shows DET curves for the *ResNetSE34V2* model evaluated on the three evaluation sets. In an operating range of FPR between 0.1% and 5% the DET curves of *VoxCeleb 1-test* and *VoxCeleb 1-E* overlap. This indicates that the randomly paired utterances of these two sets result in similar evaluations for a reasonable range of operating conditions. However, the DET curve of *VoxCeleb 1-test* is irregular, confirming that this evaluation set is small and indicating that the evaluation is unreliable. The DET curve of *VoxCeleb 1-H* is smooth, indicating that the size of the evaluation set is sufficient. However, this DET curve lies significantly above those of the other two evaluation sets. At the same FPR, the model thus has a larger FNR when evaluated on this evaluation set. This shows that the performance of speaker verification models is highly susceptible to the size and difficulty of the evaluation set.

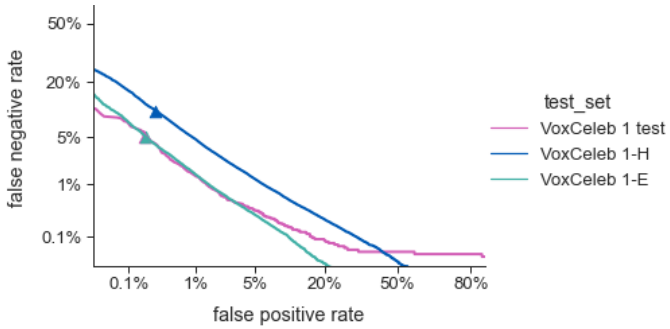


Figure 5.4: Evaluation bias in the three VoxCeleb 1 evaluation sets with *ResNetSE34V2*

5. Critical reflection on evaluation datasets

The VoxCeleb 1 evaluation sets are not representative and inadequately capture many application contexts, especially those common in voice-activated Edge AI. They present an oversimplified view of real-life application scenarios, which further limits their ability to evaluate realistic application settings. Using these evaluation sets to evaluate the performance of speaker verification models is bound to lead to an overestimate of model performance for some speaker groups, while inadequately evaluating model performance for other groups. This introduces bias into the evaluation. The evaluation sets can be improved by developing more robust approaches for generating representative evaluation sets with controlled utterance pairings from VoxCeleb 1. Moreover, evaluation sets generated from VoxCeleb 1 should be used along other datasets that better capture application contexts appropriate to real use cases.

Group design

Capture the rationale for choosing attributes that define groups and for deciding when enough different groups have been evaluated.

The pattern is used as is, and includes the following sections:

1. Group attributes
2. Rationale for group design

---

1. Group attributes

The bias evaluation considers all intersectional speaker groups that can be constructed from gender and nationality attributes captured in the VoxCeleb 1 metadata. These groups are not exhaustive, put present a reasonable starting point as no prior bias evaluations of this speaker verification benchmark have been done.

2. Rationale for group design

Gender is a protected attribute in many domains and countries. Moreover, it is well established that gender affects the performance of speaker recognition [108]. In the

past, some speaker recognition challenges have reported performance separately for male and female speakers. The submissions to the 2013 SRE in Mobile Environments, for example, made it clear that disparate error rates ought to be a cause of concern in speaker recognition systems [146]. Of 12 submissions to the challenge, all submitted systems performed worse for females than for males on the evaluation set. On average the error rate for females was 49.35% greater than for males. In later works the discrepancy between female and male speakers is still evident and reported, but remains unquestioned and unaddressed [221]. Historically, a common approach to avoid gender-based bias was to develop separate models for female and male speakers [151]. DNN-based speaker verification system no longer do this.

Evaluating speaker verification performance disparities due to nationality is limited (see the discussion on the *Metadata approximation* pattern in Section 5.3.2) but has merits. Discrimination based on national origin can have legal consequences, for instance Title VII of the Civil Rights Act of 1964 prohibits employment discrimination based on national origin in the United States.

Metadata approximation

Approximate metadata labels from other features in the dataset.

The pattern is reframed as a question: *Are approximated metadata labels in VoxCeleb 1 a source of bias in the evaluation?* The pattern analysis includes the following sections:

1. Approximation approach
2. Consequences for evaluation
3. Critical reflection on metadata approximation

---

1. Approximation approach

The VoxCeleb 1 dataset creators inferred nationality labels from speakers' countries of citizenship, as obtained from Wikipedia. Their underlying motivation for doing this was to assign a label that is indicative of a speaker's accent, and they considered citizenship as a better proxy for accent than ethnicity [209]. While nationality is a reasonable axis of analysis for bias, conflating nationality and accent is problematic. People with the same citizenship can speak the same language with different accents. Likewise, many countries have citizens speaking different languages, even if they have the same nationality. For example, India has 7 languages with more than 50 million first language speakers each [323], and many more languages spoken by several million people.

The VGGFace 1 dataset [222] provided the candidate speakers and associated gender metadata labels for VoxCeleb 1. VGGFace 1 obtained the labels from the Internet Movie Database celebrity list. Only binary gender categories, namely male and female,

have been recorded. Many concerns about gender labelling in face analysis technologies have been pointed out in prior research [257]. These concerns critique the encoding of gender into binary labels in ML datasets that favour the use of gender as a demographic imposed by society and the behaviours or appearances associated with societal gender stereotypes. In speaker recognition systems, binary gender labels are as limited as in face analysis systems. Moreover, even people that fit binary gender categories can have voice attributes that cannot be clearly ascribed to one gender or the other. Simply replacing a binary gender classification with more categories is not a recommended alternative. Even if it were possible to produce accurate labels, they might help to mitigate bias in speaker verification only while increasing potential for harm in other ways, for example through voice-based gender classification enabled targeting.

2. Consequences for evaluation

While metadata labels are not used for making predictions, they are used to make judgements about the representativeness of datasets and inform group design. This gives them a central role in group-based bias evaluations. If no metadata is available, approximating labels can be a useful tool for enabling a bias evaluation. However, if not considered carefully, this pattern can amplify representation bias and stereotyping.

3. Critical reflection on metadata approximation

In speech processing systems, like speaker recognition, inferring metadata labels with a direct relation to voice attributes, such as pitch, rate of speech or fundamental frequency, offers an alternative approach to labelling speech data. Voice attributes can then be correlated to demographic attributes in applications where this is important to assess discrimination and fairness.

Ground-truth label distribution across groups

Ensure that ground-truth labels are balanced across groups.

The pattern is reframed as an analysis instruction: *Analyse the label distribution across groups to ensure that ground-truth labels are balanced across groups.*

By design all three evaluation sets contain the same count of utterance pairs from same speakers and different speakers across all groups. This is standard practice in speaker verification evaluations and means that the evaluation sets are balanced across true positive (same speaker) and true negative (different speaker) labels.

Data quantity distribution across groups

Ensure that all groups are represented sufficiently in the data.

The pattern is reframed as an analysis instruction: *Analyse the data distribution across groups to ensure that all groups are represented sufficiently.* The pattern analysis includes the following sections:

1. Levels of analysis
2. Data distribution across speakers and utterances
3. Consequences for evaluation

---

1. Levels of analysis

Speaker verification evaluation is done on utterance pairs coming from the same and from different speakers. Representativeness is thus important on the speaker and on the utterance level. On the speaker level it ensures that the evaluation set includes a variety of speakers. On the utterance level representativeness is necessary to ensure that sufficient speech samples are included for each individual speaker. A significant mismatch in representativeness between the speaker and utterance level is undesirable. If the proportion of samples from a group in a dataset is higher on the speaker level than the utterance level, this indicates that individual speakers of that group have a low utterance count per speaker and may be insufficiently represented. Evaluation sets that underrepresent individual speakers can lead to unreliable evaluations for affected group due to small sample sizes. Conversely, if the proportion of a group is lower on the speaker level than the utterance level, speakers in that group have a higher utterance count per speaker, which indicates that individual speakers may be overrepresented.

2. Data distribution across speakers and utterances

The distribution of speakers in the VoxCeleb 1 dataset is skewed towards males, US nationals and anglophone countries, as shown in Figure 5.5. This skewed distribution in the dataset is carried over into the evaluation sets, as shown in Table 5.3. In the *VoxCeleb 1-test*, *-E* and *-H* evaluation sets the proportion of female speakers is, respectively, 61% (38/62), 82% (45/55) and 79% (44/56) that of male speakers. When considering utterances of female speakers, the proportion of females decreases to 42% (29.5/70.5), 72% (41.8/58.2) and 70% (41.1/58.9) that of males across the three sets. Thus, not only do the evaluation sets contain fewer female speakers, they also contain fewer utterances for each female speaker, which makes the evaluation less reliable for females. For nationality metadata, utterance level representativeness slightly improves for the three evaluation sets, as the proportion of dominant nationality (i.e. US) utterances decreases.

3. Consequences for evaluation

Unbalanced and insufficiently represented groups in evaluation sets provide an inadequate understanding of the real capabilities of speaker verification for a diverse population of people. Groups with the most speakers have the least variability in performance, and their performance aligns the closest with average performance. On the

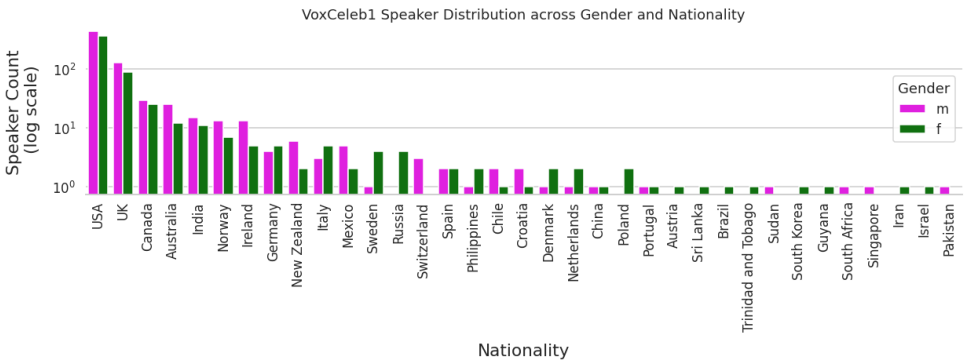


Figure 5.5: VoxCeleb 1 Dataset speaker distribution across gender and nationality.

other hand, performance is highly variable for groups with small amounts of speakers. Empirically, this statement will be validated in the next subsection in Figure 5.7, where the DET curves of small groups, like Italian, German and Irish speakers, are jagged and variable, while DET curves of US and UK speakers are smooth and reliable. In addition to influencing the quality of the evaluation, evaluation sets with insufficient representativeness also orientate the development of speaker verification technology towards the groups of people that are most represented.

New pattern: Data source

Ensure that the data source is legitimate and that the dataset contains no unforeseen bias.

The pattern is reframed as an analysis instruction: *Trace the data creation pipeline from start to finish to ensure that the data source is legitimate and that the dataset contains no unforeseen bias.* The pattern analysis includes the following sections:

1. Automated Data Creation Pipeline
2. Consequences for evaluation

---

1. Automated Data Creation Pipeline

The VoxCeleb 1 dataset was generated with a fully automated data processing pipeline from open-source audio-visual media [207]. The candidate speakers for the dataset were sourced from the VGGFace 1 dataset [222], which includes the intersection of the most searched names in the Freebase knowledge graph and Internet Movie Database (IMDB). After searching and downloading video clips for identified celebrities, further processing was done to track faces, identify active speakers and verify the speaker's identity using a HOG-based face detector [150], Sync-Net [42] and VGG Face CNN [269] respectively. If the face of a speaker was correctly identified, the clip was included in the dataset.

2. Consequences for evaluation

The inclusion criteria of candidate celebrity identities in VGGFace 1 reinforces popularity bias in search results. This bias is directly transferred to VoxCeleb 1. Moreover, the automated data generation process of VoxCeleb 1 propagates known bias in facial recognition technologies, which have been shown to have inferior predictive performance for people of colour and in particular women of colour [30, 238, 239]. The data processing pipeline thus transmits bias in facial recognition systems into the speaker verification domain, as failures in the former will result in speaker exclusion from VoxCeleb 1.

5.3.3. Using the Bias Measurement Recipe

Bias Measurement Recipe

- o 1 Types of harm
- o 2 Fairness assumptions
- \+ 3 Group design
- o 4 Metadata approximation
- \+ 5 Disaggregated base metrics
- o 6 Group-based bias measure
  - o Visualising metrics and measures
  - o ~~Meta-measure~~

The *Bias Measurement* recipe contains patterns to support the quantification of bias in ML models. The recipe, as introduced in Chapter 4, includes eight patterns. Together, the patterns enable a transparent and reproducible quantitative assessment of bias that states assumptions about the evaluation upfront. This section validates six of the patterns in the recipe by using them to quantify bias in the VoxCeleb SRC baseline models. The recipe is used in proactive mode to set up a bias evaluation, and patterns are applied in the order specified in the recipe, as they are dependent on each other. The optional[2] *Meta-measure* pattern is not used in the analysis as bias is not compared across models in this study, and a meta-measure is thus not relevant.

Types of harm

Identify context and application specific risks and consequent harm that a biased model can lead to.

The pattern is used as is, and includes the following sections:

1. Types of errors
2. Taxonomy of risks
3. Discussion of consequences

---

1. Types of errors

Speaker verification systems produce two types of errors, false positives (FP) and false negatives (FN) (see Section 5.2.2 for an overview of speaker verification evaluation).

[2] indicated in the recipe with the prefix 'o'

The FP error rate (FPR) and FN error rate (FNR) present a trade-off in speaker verification systems. Any speaker verification system with a low FPR will have a higher FNR. FP errors in biometric authentication applications *grant unauthorised speakers access* to a system and thus pose a security risk. Historically speaker verification research has been funded by governments to advance biometric identification and authentication applications for intelligence, defense and justice objectives [101]. Speaker verification evaluations have thus focused on obtaining low FPR.

2. Taxonomy of risks

Speaker verification is used in a wide range of voice-based applications, ranging from voice assistants on smart speakers and mobile phones to call centers. In many applications a low FPR remains necessary to ensure system security. However, security is not the only thing that is at stake when speaker verification systems fail. Performance errors have different consequences that result in a variety of risks for different applications. Table 5.4 presents a taxonomy of error types, consequences, risks and corresponding applications in speaker verification systems.

Error type	Consequence	Risk	Example application
FP	mistaken identity	False accusation	criminal justice system
FP	mistaken identity	Identity theft	mobile banking
FP	unsolicited data collection	Compromised privacy	voice assistant
FP	unauthorised access	Security breech	mobile phone
FN	access denied	Quality-of-service harm	voice assistant
FN	access denied	Allocation harm	proof-of-life verification, workforce monitoring

Table 5.4: Types of harm that can result from FP and FN errors of speaker verification systems

3. Discussion of consequences

For example, in voice assistants positive classifications trigger voice data to be sent to service providers for downstream processing [258]. FP errors thus compromise privacy. When used in forensic applications in the criminal justice system, FP errors can lead to mistaken identities and false accusations, potentially amplifying existing systemic bias [139]. FN errors also carry consequences. FN errors affect usability and can lead to a denial of service from voice-based user interfaces, resulting in a quality-of-service harm. The more critical the service, the greater the risk of harm associated with FN errors. Consider, for example, a speaker verification system used for proof-of-life verification of pensioners [204]. As long as the system is able to identify a pensioner correctly, it relieves the elderly from needing to travel to a government office to collect their pension. This saves them time, money and physical strain. If the system however has a high FNR and fails more frequently to identify certain pensioners correctly, it will subject these individuals to a greater burden of travel. The potential quality-of-service harms that can result from FN errors then turn into allocation harms.

Fairness assumptions

Make assumptions explicit with an upfront statement about that which is believed to make a system fair.

The pattern is used as is, and includes the following sections:

1. Fairness considerations
2. Statement of assumptions

---

1. Fairness considerations

Speaker verification systems can be active (i.e. invoked by the speaker) or passive (i.e. running in the background and not explicitly invoked), used with or without consent of the speaker, and intended to advance the interests of the speaker or not. Assessing whether a speaker verification system is fair is thus a complex matter. For example, a passive speaker verification system in a call center, used without informed consent of the speaker and with the intention to demote the speaker's customer query as they are not deemed a valuable customer, would be difficult to motivate as being fair.

2. Statement of assumptions

For the purpose of evaluating bias in the VoxCeleb SRC benchmark, speaker verification system can be either active or passive, but are assumed to be used with consent of the speaker to advance their interests. Voice assistants in smart speakers contain such speaker verification systems to protect the device from intruders. In these applications, a fair system should then perform equally for all users. In this evaluation this is interpreted as the system having equal base metrics for all users.

Group design

See Section 5.3.2.

Metadata approximation

See Section 5.3.2.

Disaggregated base metrics

Calculate performance metrics across groups.

The pattern is used as is, and includes the following sections:

1. Overview of metrics
2. Model evaluation across groups
3. Critical reflection on metrics

---

1. Overview of metrics

The two main base metrics used in speaker verification benchmarks, including the VoxCeleb SRC, are the equal error rate (EER) and the minimum value of the detection cost function (DCF) $min\ C_{Det}$ (see Table 5.1). Table 5.5 disaggregates these base metrics across intersectional gender and nationality groups for the two models evaluated on VoxCeleb 1-H. $min\ C_{Det}$ cannot be calculated directly for each group. Instead, the table shows $C_{Det}(\theta_{@\ overall\ min})$ which is the detection cost for the group at the threshold θ where the overall DCF is at its minimum. The table also shows the corresponding error rates $FPR(\theta)$, and $FNR(\theta)$ for $\theta_{@\ overall\ min}$.

	Speaker	ResNetSE34V2				ResNetSE34L			
Group	**Count**	EER	$C_{Det}(\theta)$	$FPR(\theta)$	$FNR(\theta)$	EER	$C_{Det}(\theta)$	$FPR(\theta)$	$FNR(\theta)$
New Zealand (m)	6	1.4381	0.0052	0.0006	0.0928	2.9314	0.0080	0.0000	0.1602
Ireland (f)	5	1.5326	0.0055	0.0010	0.0920	1.5326	0.0084	0.0010	0.1504
USA (m)	431	1.8792	0.0065	0.0011	*0.1094**	3.3780	0.0118	0.0020	*0.1977**
USA (f)	368	2.0076	0.0071	0.0022	0.1016	4.0726	0.0141	*0.0062**	0.1644
UK (m)	127	2.2148	0.0074	*0.0037**	0.0764	3.7054	0.0115	0.0032	0.1693
India (m)	15	2.2295	*0.0095**	*0.0071**	0.0544	*4.5455**	*0.0165**	*0.0104**	0.1332
Ireland (m)	13	*2.4770**	*0.0081**	0.0018	*0.1277**	3.8560	0.0128	0.0008	*0.2416**
Canada (m)	29	*2.4849**	0.0057	0.0005	0.1028	*4.5679**	0.0127	0.0018	*0.2183**
Australia (f)	12	*2.5241**	*0.0089**	*0.0059**	0.0657	4.0089	0.0130	*0.0071**	0.1251
UK (f)	88	*2.5840**	*0.0113**	*0.0083**	0.0675	*5.0241**	*0.0192**	*0.0130**	0.1373
Mexico (m)	5	*2.7434**	0.0045	0.0000	0.0894	4.3363	*0.0174**	0.0000	*0.3478**
Australia (m)	12	*2.8791**	0.0070	0.0012	*0.1175**	*4.8544**	*0.0148**	0.0023	*0.2506**
Canada (f)	25	*3.6707**	*0.0112**	*0.0033**	*0.1613**	*6.8595**	*0.0185**	*0.0065**	*0.2477**
Italy (f)	5	*4.0219**	*0.0138**	*0.0110**	0.0678	*6.2609**	*0.0209**	*0.0146**	0.1409
Norway (f)	7	*4.8797**	*0.0114**	0.0007	*0.2152**	*9.2246*	*0.0232**	*0.0080**	*0.3108**
India (f)	11	*5.6259**	*0.0200**	*0.0138**	*0.1367**	*10.1500**	*0.0466**	*0.0391**	*0.1878**
Germany (f)	5	*6.8471**	*0.0104**	0.0016	*0.1768**	*11.6242**	*0.0217**	*0.0104**	*0.2373**
Norway (m)	13	*7.5953**	*0.0199**	*0.0065**	*0.2760**	*10.6745**	*0.0298**	*0.0094**	*0.4176**
average		**2.4023**	**0.0077**	**0.0027**	**0.1036**	**4.3733**	**0.0142**	**0.0051**	**0.1857**

Table 5.5: ResNetSE34V2 and ResNetSE34L model performance evaluated on VoxCeleb 1-H, disaggregated across base metrics (EER, $C_{Det}(\theta)$, $FPR(\theta)$, $FNR(\theta)$, for $\theta_{@\ overall\ min}$. Starred metric *values** indicate groups that perform worse than average for that metric and model.

2. Model evaluation across groups

From the results in Table 5.5 it is evident that significant group-based differences exist in the performance of the models. Only 6 out of 18 groups have an EER better than average when using ResNetSE34V2. The worst performing group, Norwegian males, has an EER that is more than 3 times higher than average, and a detection cost that is more than 2.5 times higher. Correspondingly, when calibrated to the average minimum threshold, the likelihood that speakers of this group will be subjected to FP and FN errors is around 2.5 times greater than average. Overall, ResNetSE34L performs worse than ResNetSE34V2. This is to be expected, as the former is a smaller model with fewer parameters. When considering the disaggregated EER of ResNetSE34L, half the groups perform better and half perform worse than average. Examining $C_{Det}(\theta)$, $FPR(\theta)$ and $FNR(\theta)$ across groups for this model shows that performance difference

between groups do not diminish. In fact, Indian females, who have the highest $C_{Det}(\theta)$, are 7.6 times more likely than average to experience FP errors. Their high FPR is accompanied by a relatively low FNR that is marginally higher than average. Norwegian males, who have the highest FNR at 2.2 times greater than average, still have a FPR that is 1.8 times greater than average.

3. Critical reflection on metrics

While the EER can be useful to compare models between each other, it presents an oversimplified view of model performance for real applications. The metric cannot weight FP and FN errors differently, yet most speaker verification applications strongly favour either a low FPR or a low FNR. The NIST SREs do not promote the use of the EER for evaluation for this reason [101]. The $min\ C_{Det}$ and $C_{Det}(\theta)$ can weight the FPR and FNR, but have their own shortcomings. Firstly, the DCF has been updated over the years, and different versions of the metric are in use. This is impractical for consistent evaluation of applications across time. Secondly, the cost function is only useful if the FPR and FNR weighting reflect the requirements of the application. Determining appropriate weights is a difficult and normative design decision. Competitions like the VoxCeleb SRC typically do not adjust the weights, which once again oversimplifies real-life evaluation scenarios and limits the evaluation to a single threshold value.

Visualising metrics and measures

Plot metrics and measures across groups to visually examine performance disparities.

The pattern is used as is, and includes the following sections:

1. Visual representation of performance
2. Visual performance analysis across *Gender* groups
3. Visual performance analysis across *Nationality + Gender* groups

---

1. Visual representation of performance

DET curves provide a view on the performance of speaker verification models across calibration thresholds. Provided that sufficient utterance pairs have been evaluated, they visualise the theoretical performance boundary of a model across groups. This analysis visualises DET curves for *ResNetSE34V2* evaluated on *VoxCeleb 1-H*. The other two evaluation sets and the *ResNetSE34L* model exhibit similar trends as those shown in this analysis. For conciseness these visualisations are not included.

2. Visual performance analysis across *Gender* groups

Figure 5.6 visualises the DET curves for female (left) and male (right) groups of 11 nationalities. The dotted black DET curve shows the overall performance across all groups. DET curves above the dotted line will typically have worse than average EER and $C_{Det}(\theta)$ base metrics, while DET curves below the dotted line will generally perform better than average. The visualisation makes it easy to see that the DET curves

of females mostly lie above the overall DET curve, while those of males lie on or below it. The model will thus perform worse than average for females, and better for males at most operating thresholds. The triangle markers show the FPR and FNR when the model is calibrated to $\theta_{@\ overall\ min}$, the threshold that minimises the overall system DCF. The markers for all groups are dispersed, which highlights that calibrating the model to $\theta_{@\ overall\ min}$ will result in significant error rate differences across groups.

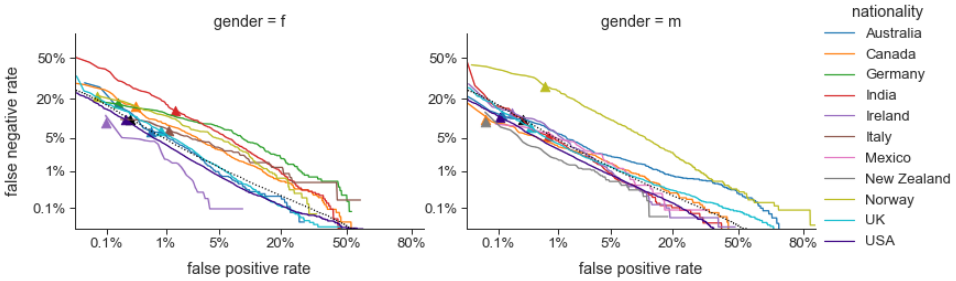


Figure 5.6: Disaggregated DET curves by gender and nationality for *ResNetSE34V2* evaluated on *VoxCeleb 1-H*. The dotted black line captures overall performance, triangle markers indicate the FPR and FNR at the calibration threshold $\theta_{@\ overall\ min}$

3. Visual performance analysis across *Nationality + Gender* groups

Figure 5.7 disaggregates the DET curves to show disparate performance across intersectional nationality and gender groups. The figure visually confirms the performance results in Table 5.5. Closer inspection of the DET curves clearly shows that the model is fit to the dominant population in the training data, US speakers. The curves of male and female speakers from the US are smooth and they lie close to the overall curve. Contrast this against the DET curve of female Indian speakers, which lies far above the overall curve. Irrespective of the threshold, the model will perform worse than average for this group. Female and male speakers from the US retain the average FNR and have a better than average FPR when the model is calibrated to the threshold $\theta_{@\ overall\ min}$ (triangles). For other groups, like UK females and Indian females and males, either the FPR or the FNR deteriorates significantly when the model is calibrated to this value.

Group-based bias measure

Compare base metrics across groups or between groups and overall performance to assess whether different groups are treated equally.

The pattern is used as is, and includes the following sections:

1. Overview of bias measure
2. Measuring bias across groups

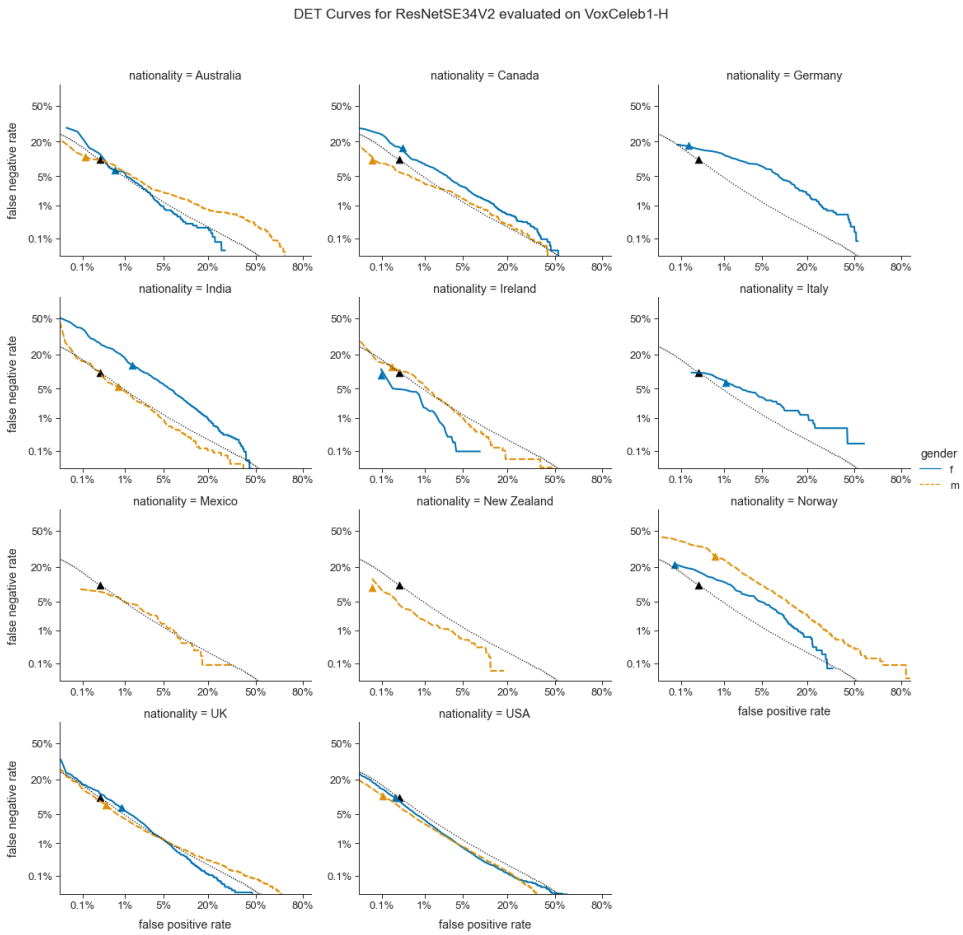


Figure 5.7: DET curves disaggregated by nationality and gender for *ResNetSE34V2* evaluated on *VoxCeleb 1-H*.

1. Overview of bias measures

As a final step in the evaluation, the group-to-overall ratio is used as bias measure to compare the group-based performance differences measured by the base metrics. The group-to-overall ratio calculates the relation between the group and the overall base metric, and is calculated for the EER, $C_{Det}(\theta)$ (as defined in Equation 5.1), $FPR(\theta)$ and $FNR(\theta)$ for $\theta_{@\ overall\ min}$. Equation 5.2 is a generic equation for calculating the bias measure for group g, with a specific example of how to calculate the measure for the $C_{Det}(\theta)$ base metric.

$$bias_g^{base\ metric} = \frac{base\ metric_{spk_g}}{base\ metric_{overall}} \tag{5.2}$$

$$bias_g^{C_{Det}(\theta)} = \frac{C_{Det}(\theta_{@\ overall\ min})_g}{min\ C_{Det}}$$

If the bias measure is greater than 1, the group performance is worse than the overall performance, and the speaker verification system is prejudiced against that group. Conversely, if the bias measure is less than 1, the group is favoured. If the ratio is exactly 1, the group's performance equals overall performance.

2. Measuring bias across groups

Table 5.6 shows the values of the bias measures for all groups and base metrics. Using the bias measure makes it easy to see which groups perform better and which perform worse than average. Moreover, the ratio-based bias measures make performance differences comparable across base metrics, even if the metrics are of different orders of magnitude. For example, it is now easy to see that Italian female speakers have a worse than average EER, $C_{Det}(\theta)$ and FPR for both models, but that their FNR is better than overall. Norwegian males, Canadian and Indian females, on the other hand, have worse than overall performance for all metrics. While the bias measures calculated from $C_{Det}(\theta)$ for Italian and Canadian females lie relatively close, comparing the bias measures calculated from the FPR immediately shows that these two groups will be subjected to different risks, with Italian females much more likely to experience FP errors, and Canadian females somewhat more likely to experience FP and FN errors.

	ResNetSE34V2				ResNetSE34L			
Group	$bias_g^{EER}$	$bias_g^{C_{Det}(\theta)}$	$bias_g^{FPR(\theta)}$	$bias_g^{FNR(\theta)}$	$bias_g^{EER}$	$bias_g^{C_{Det}(\theta)}$	$bias_g^{FPR(\theta)}$	$bias_g^{FNR(\theta)}$
New Zealand (m)	0.5986	0.6668	0.2047	0.8957	0.6703	0.5656	0.0000	0.8628
Ireland (f)	0.6380	0.7109	0.3546	0.8874	0.3504	0.5952	0.1866	0.8098
USA (m)	0.7823	0.8357	0.3924	1.0554	0.7724	0.8320	0.3897	1.0644
USA (f)	0.8357	0.9224	0.8060	0.9801	0.9312	0.9967	1.2091	0.8852
UK (m)	0.9219	0.9523	1.3866	0.7372	0.8473	0.8090	0.6138	0.9116
India (m)	0.9281	1.2200	2.6230	0.5250	1.0394	1.1657	2.0197	0.7170
Ireland (m)	1.0311	1.0432	0.6617	1.2322	0.8817	0.9042	0.1492	1.3008
Canada (m)	1.0344	0.7304	0.2029	0.9916	1.0445	0.8932	0.3559	1.1754
Australia (f)	1.0507	1.1523	2.1985	0.6340	0.9167	0.9147	1.3737	0.6736
UK (f)	1.0756	1.4558	3.0806	0.6509	1.1488	1.3566	2.5314	0.7394
Mexico (m)	1.1420	0.5768	0.0000	0.8626	0.9915	1.2278	0.0000	1.8728
Australia (m)	1.1985	0.9020	0.4337	1.1340	1.1100	1.0419	0.4564	1.3494
Canada (f)	1.5280	1.4501	1.2353	1.5565	1.5685	1.3096	1.2638	1.3337
Italy (f)	1.6742	1.7827	4.0604	0.6545	1.4316	1.4783	2.8486	0.7586
Norway (f)	2.0313	1.4711	0.2474	2.0771	2.1093	1.6354	1.5623	1.6738
India (f)	2.3419	2.5766	5.1160	1.3188	2.3209	3.2869	7.6193	1.0111
Germany (f)	2.8503	1.3359	0.5894	1.7057	2.6580	1.5319	2.0159	1.2776
Norway (m)	3.1617	2.5720	2.3882	2.6630	2.4408	2.1037	1.8278	2.2487

Table 5.6: ResNetSE34V2 and ResNetSE34L bias measures evaluated on VoxCeleb 1-H. Values greater than 1 indicate that the speaker verification system is biased against that group.

5.4. Mitigating Benchmark Dataset Pitfalls

Section 5.3.2 used the *Benchmark Dataset* recipe to highlight various shortcomings of using the VoxCeleb 1 evaluation sets for evaluating speaker verification systems. One of the shortcomings is that the three benchmark evaluation sets of VoxCeleb 1 result in evaluations of varying difficulty and variable reliability across speaker groups. The small evaluation set, VoxCeleb 1-test, and speaker groups with few data samples have the greatest variablility in their evaluation results. This section investigates this further, and shows empirically that the quantity and difficulty of same and different utterance pairs across speaker groups affects evaluation outcomes. Based on these insights, criteria for constructing more representative and robust speaker verification evaluation sets from VoxCeleb 1 are proposed. The section shows that these criteria can be considered as instantiations of patterns in the *Benchmark Dataset* recipe in the speaker verification domain. Framing the criteria as pattern instantiations makes it possible to clearly communicate their contribution to bias mitigation.

The section starts with an outline of design requirements for robust evaluation sets, and then proposes an algorithm for generating such evaluation sets. This is followed by an empirical analysis of the evaluation sets that are constructed using this approach. The section concludes by capturing insights gained from the analysis as criteria for constructing evaluation sets. The ideas captured by the criteria are generalised and mapped to patterns. Finally, the section proposes a new dataset pattern, *Data quality distribution across groups*, to extend the *Benchmark Dataset* recipe. The work is based on research published in [130].

5.4.1. Evaluation Set Requirements

This study is motivated by the analysis of the *Evaluation datasets* and *Data quantity distribution across groups* patterns in Section 5.3.2 and investigates requirements for representativeness and sufficiency in speaker verification evaluation sets. The study focuses on the often-times neglected analysis level of utterance pairs and aims to answer two key questions that address sufficiency and representativeness:

Q1: How many utterance pairs are needed per speaker for a robust speaker verification evaluation?

Q2: How should utterance pairs be constructed to support a robust evaluation?

Terminology and notation

A *speaker* (S^i) is a unique person with at least one speech *utterance* (u^i) in the set of all *evaluation utterances* ($\mathcal{U}$). $\mathcal{U}^i$ is the subset of evaluation utterances of speaker S^i and $S = \{S^1 ... S^K\}$ is the set of all speakers in $\mathcal{U}$. An *evaluation set* ($\mathcal{D}$) is constructed by generating utterance pairs $\{u_a, u_b\}$ from the evaluation utterances in $\mathcal{U}$. Conventionally u_a is the enrollment utterance, and u_b is the test utterance. Practically, the order of utterance pairs does not matter in speaker verification evaluation. A *same speaker*

pair $\{u_a^1, u_b^1\}$ has the enrollment and test utterances drawn from $\mathcal{U}^1$, while a *different speaker* pair $\{u_a^1, u_a^2\}$ has the enrollment and test utterances drawn from two different speakers, ie from $\mathcal{U}^1$ and $\mathcal{U}^2$.

Design requirements

The objective of creating a robust speaker verification evaluation set, $\mathcal{D}$, is then to:

1. Offer an equivalent evaluation across all speakers in S
2. Generate same speaker and different speaker utterance pairs that are reflective of real-life usage scenarios
3. Conduct an evaluation that is robust to perturbations in utterance pairings (e.g. if $\{u_a^1, u_a^2\}$ is substituted with $\{u_b^1, u_f^3\}$ the false positive error rate should not change)

5.4.2. Generating Robust Evaluation Sets

To offer an equivalent evaluation across speakers and speaker groups, evaluation conditions must be similar for all speakers. This means that evaluations must be of similar difficulty, and that sufficient utterance pairs are evaluated for all speakers to draw reliable conclusions. Next, I present a taxonomy for grading the difficulty of utterance pairs, and an algorithm that takes these requirements into consideration to generate robust evaluation sets.

Difficulty Grading of Utterance Pairs

Speaker verification is a comparative task that assesses the similarity between an enrollment utterance and a test utterance based on the difference between embeddings of the utterances. Naturally, system performance is thus impacted by the utterance pairs that are being compared. However, not all utterance pairs are born equal. Utterances from speakers with very different voice attributes are easier to tell apart, and make for easier *different speaker* pairs. Conversely, *same speaker* pairs are more difficult to identify if the utterances have different attributes. Consequently, an evaluation of *same speaker* utterances is easier if the pairs are more similar, and harder if utterance pairs are more different. For different speakers the opposite is true: similar utterance pairs are harder, and different utterance pairs are easier to classify. A robust speaker verification evaluation should test utterance pairs that represent likely scenarios that a system will encounter in applications, and should evaluate pairs of appropriate difficulty to test the limits of the system. Prior to this study no grading scheme of utterance pairs existed, which lead us to create the typology in Table 5.7 for grading the difficulty of utterance pairs.

Speaker demographics, channel and environmental attributes are some of the factors that determine the similarity of utterance pairs. The typology uses available metadata to grade the difficulty of utterance pairs based on speakers' gender, nationality, recording channel and background noise. It assigns four difficulty categories: trivial,

Utterance Pairs	Difficulty	Same Gender	Same Nationality	Same Channel	Same Noise
Same Speaker	trivial	-	-	Yes	Yes
	medium	-	-	No	n.k.
Different Speakers	trivial	No	No	-	n.k.
	easy	No	Yes	-	n.k.
	medium	Yes	No	-	n.k.
	hard	Yes	Yes	-	n.k.

Table 5.7: Grading of utterance pairs (n.k. = not known)

easy, medium and hard for same and different speaker pairs. For example, comparing two utterances of the same speaker from the same video clip implies that both the recording channel and the background noise are highly likely to be the same. Such a *same speaker* comparison is thus trivial. Or male and female speakers with different nationalities and thus different accents are likely to have very different voice profiles, making this *different speaker* comparison trivial.

Algorithm Design

Taking the considerations discussed above and the dataset requirements into account, Algorithm 1 generates the hardest possible evaluation set, $\mathcal{D}$, from the set of available utterances, $\mathcal{U}$. To ensure that the evaluation is robust to perturbations in utterance pairings, the random seed for selecting same and different speaker utterance pairs can be changed to generate similar evaluation sets with different utterance pairings. Speaker verification evaluation outcomes can then be compared across several equivalent sets.

5.4.3. Empirical Validation

The proposed algorithm is empirically validated by using it to generate speaker verification evaluation sets from the VoxCeleb 1 dataset. This section describes the experiment setup and the baseline evaluation set against which the algorithmically generated evaluation sets are compared.

Experiment Setup

The empirical experiments evaluate the pre-trained *ResNetSE34V2* model described in Section 5.3.1 and Table 5.2. The model is used in a speaker verification inference pipeline to classify utterance pairs in evaluation set $\mathcal{D}$ as being from the same or from different speakers. The only variable that is considered in the experiments are the evaluation sets. The evaluation sets were generated from VoxCeleb 1 with Algorithm 1. Different speaker pairs were constructed from speakers with the same gender and nationality.

Algorithm 1 Speaker Verification Evaluation Set Generation Algorithm

$S \leftarrow$ set of all speakers
$i \leftarrow$ unique speaker
$a, b \leftarrow$ recording instances
$u \leftarrow$ utterance
$\mathcal{U} \leftarrow$ set of all utterances
$n \leftarrow$ count of *same* or *different speaker* utterance pairs/speaker
$r \leftarrow$ random seed
$\mathcal{D} \leftarrow$ new evaluation set

for S^i in S **do**
  $\mathcal{U}^i \leftarrow$ subset of u^i
  $\mathcal{U}^j \leftarrow$ subset of u^j ▷ $i \neq j$; S^i, S_j *same group*
  $\mathcal{D}^i_{same} \leftarrow$ new list of *same speaker* utterance pairs
  $\mathcal{D}^i_{diff} \leftarrow$ new list of *different speaker* utterance pairs

  for u^i_a, u^i_b in $\mathcal{U}^i$ **do** ▷ ***same speaker pairs***
    if recording$_a$ is not recording$_b$ **then** ▷ *difficulty: medium*
      $\mathcal{D}^i_{same}$ append $\{u^i_a, u^i_b\}$
    else
      pass
    end if
  end for

  $\mathcal{D}^i_{same} \leftarrow$ randomly select n pairs from $\mathcal{D}^i_{same}$ with seed r
  $\mathcal{U}^i_n \leftarrow$ randomly select n u from $\mathcal{U}^i$ with seed r (replace ok)
  $\mathcal{U}^j_n \leftarrow$ randomly select n u from $\mathcal{U}^j$ with seed r
  $x \leftarrow 0$

  while $x < n$ **do**
    for u^i_x, u^j_x in $\mathcal{U}^i_n, \mathcal{U}^j_n$ **do** ▷ ***different speaker pairs***
      $\mathcal{D}^i_{diff}$ append $\{u^i_x, u^j_x\}$
    end for
  end while

  $\mathcal{D}$ append $\mathcal{D}^i_{same}$
  $\mathcal{D}$ append $\mathcal{D}^i_{diff}$
end for

The attributes that characterise evaluation sets are the speaker nationalities, the count of utterance pairs/speaker (n)[3] and the random seed with which the dataset was generated. Table 5.8 shows an overview of the variables that were chosen for the evaluation set attributes. To address Q1 and establish how many utterance pairs are sufficient for a robust evaluation, evaluation sets were generated for three nationalities with a high number of speakers. For each nationality, evaluation sets were then

[3] n is the count of same *or* different utterance pairs. As these two counts are equal, the total number of utterance pairs per speaker is $2n$.

generated with 7 different n and with 5 different random seeds, resulting in a total of 105 experiments. After analysing the results of these experiments, evaluation sets with 520 utterance pairs/speaker were generated for all nationalities that contained sufficient utterances per speaker. This resulted in 8 algorithmically generated evaluation sets for comparison against a baseline evaluation set to address Q2.

Research Question	**Nationalities** (speaker count)	**Pairs/speaker (n)**	**Random seed**
Q1	Canada (54), India (26), UK (215)	50, 100, 150, 225, 350, 450, 520	3, 6, 8, 12, 20
Q2	Canada, India, USA, Ireland, UK, Norway, Australia, Germany	520	12

Table 5.8: Overview of speaker verification evaluation experiments. n represents the count of *different speaker* utterance pairs per speaker.

Baseline Evaluation Set

To evaluate the algorithmically generated evaluation sets for Q2, they are compared against a baseline evaluation on VoxCeleb 1-H (see Section 5.3 for further details). *Different speaker* pairs in all evaluation sets, including the baseline, are constructed from speakers with the same nationality and gender and are thus categorised as hard. However, the baseline evaluation set contains trivial *same speaker* pairs, which is not the case for the algorithmically generated evaluation sets.

Table 5.9 summarises the attributes of *same speaker* pairs in VoxCeleb 1-H. The table shows that the number of speakers, the total number of utterance pairs, and the count of utterances pairs per speaker varies across nationalities. Additionally, all groups in the baseline evaluation set contain trivial *same speaker* pairs, but the proportion of trivial pairs in relation to a group's total *same speaker* pairs differs across groups. This means that some groups are evaluated under much easier conditions than others. VoxCeleb 1-H thus does not offer an equivalent evaluation across groups.

Nationality	**# Speakers**	**Pairs**	**Pairs/speaker**	**trivial**
USA	799	178122	222.9	12.9%
UK	215	53111	247.0	10.3%
Canada	54	10864	201.2	11.1%
India	26	10053	386.7	10.6%
Australia	37	8668	234.3	10.5%
Ireland	18	4960	275.6	8.5%
Norway	20	4906	245.3	10.0%
New Zealand	6	1811	301.8	10.1%
Germany	5	1256	251.2	17.0%
Mexico	5	1130	226.0	10.2%
Italy	5	571	114.2	17.0%

Table 5.9: VoxCeleb 1-H *same speaker* utterance pairs by nationality.

5.4.4. Analysis of Results

Next, I present and analyse experimental results to show how the difficulty grading and count of utterance pairs per speaker affects evaluation outcomes. First, the analysis investigates how the count of utterance pairs per speaker affects the robustness of a speaker verification evaluation when pairings are perturbed. Subsequently, the analysis contrasts evaluation outcomes between evaluation sets that carefully control the representativeness, sufficiency and difficulty of *same speaker* and *different speaker* utterance pairs, and the VoxCeleb 1-H baseline.

Effect of Utterance Pair Count

The evaluation outcomes for speakers with Canadian, Indian and UK nationalities are analysed for evaluation sets generated with 50, 100, 150, 225, 350, 450 and 520 distinct *different speaker* utterance pairs/speaker. For each utterance pair count n, five perturbed evaluation sets were generated with different random seeds. Evaluation results for the $min\ C_{Det}$ base metric are shown in Figure 5.8.

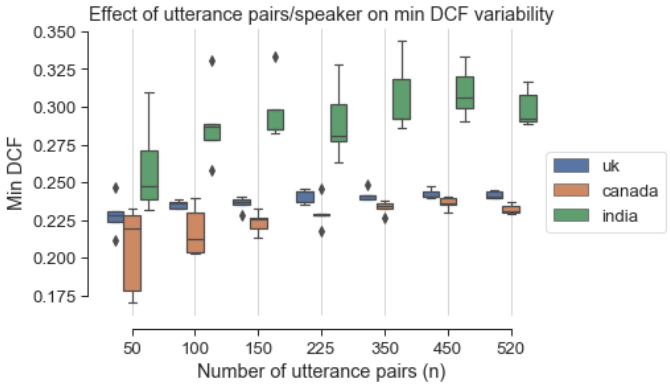


Figure 5.8: Variance in the minDCF metric for Canadian, Indian and UK speakers for evaluation sets with different counts of utterance pairs/speaker (n). Five versions of each dataset are compared. The evaluation is more robust when n is larger.

As the figure illustrates, system performance varies significantly when n is small. For example, when $n = 50$, the highest $min\ C_{Det}$ value (i.e. worst evaluation outcome) for speakers with Canadian nationality is 30% higher than the lowest $min\ C_{Det}$ value (i.e. the best evaluation outcome). Results for the EER base metric show the same trends. This performance difference can be attributed entirely to utterance pairings generated with different random seeds. In general the variance of the base metrics is greater when the count of unique speakers in a group is smaller: the UK speaker group with 215 unique speakers exhibits the lowest variance, while the Indian speaker group with only 26 unique speakers exhibits the greatest variance.

Variability is not limited to performance on the EER and $min\ C_{Det}$ base metrics,

but exists across the DET performance curve, as shown in Figure 5.9. As n increases, variability remains greatest at the end points of the curve where FP and FN error rates are the greatest or smallest. Variability is lowest in the center of the curve. This means that at very low FP rates the expected FN rate will carry significant uncertainty (and vice versa), unless n is large enough to ensure stable performance.

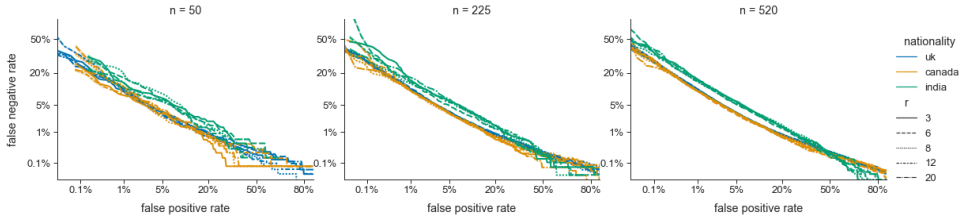


Figure 5.9: DET curves show variability in evaluation outcomes for evaluation sets with 50, 225 and 520 utterance pairs (n) for Canadian, Indian and UK speakers. For each n five datasets were generated with different random seeds.

5

Effect of Utterance Pair Difficulty Grading

Experiment 2 included speakers from 8 nationalities, with 520 utterance pairs of different speakers (1040 total pairs) per speaker. We call this dataset *VoxCeleb 1-Inclusive*. The key differences between the VoxCeleb 1-H baseline and VoxCeleb 1-Inclusive are that the latter contains an equal count of pairs/speaker across individual speakers and speaker groups, that trivial same speaker pairs have been excluded and that speakers were included if their unique utterance pair count for different speakers was greater than our selection value n. Due to the randomised utterance pair generation, the different speaker pairs in the two evaluation sets are not the same.

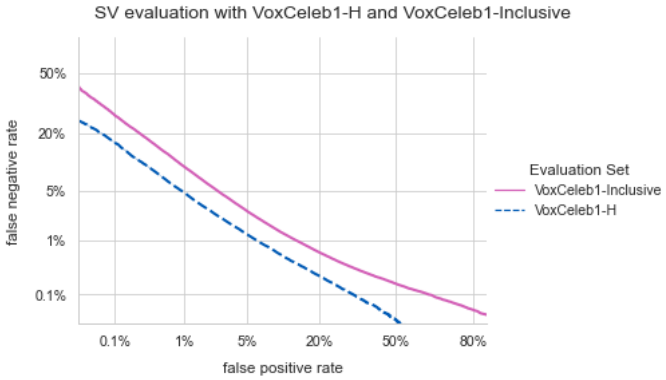


Figure 5.10: DET curves for a baseline and a carefully designed evaluation set show that the speaker verification model performs worse when sufficiency and representativeness are controlled to be equivalent across speaker groups in the evaluation set.

Figure 5.10 shows the DET performance curves for the model evaluated on the algorithmically generated evaluation set and the baseline. When evaluated on the VoxCeleb 1-H baseline, a FP rate of 1% results in a FN rate of 5%. However, when using VoxCeleb 1-Inclusive for evaluation, at the same FP rate the FN rate triples to ∼15%. VoxCeleb 1-H thus presents evaluation conditions that overestimate model performance. Model performance detriorates significanty when the evaluation exclusivly includes *same speaker* utterance pairs of medium difficulty grading, an evaluation scenario that resembles realistic operating conditions more closely.

5.4.5. Capturing Insights as Patterns

The empirical results show that the difficulty of utterance pairs impacts the outcome of speaker verification evaluations, and that randomised utterance pairings that neglect to account for the difficulty of pairs can result in significant performance variation if the utterance pair count per speaker is low. The experiments indicate that model performance deteriorates when evaluated on datasets with more difficult utterance pairs of same and different speakers. Moreover, when the number of utterance pairs per speaker is low, model performance has high variability. Evaluation outcomes become less variable (and evaluation robustness improves) as the number of utterance pairs per speaker increases.

Criteria for Speaker Verification Benchmarks

These observations can be summarised as criteria for creating speaker verification evaluation sets that consider speaker inclusion on a speaker and an utterance level. Based on the results of this study, speaker verification evaluation sets should:

1. Have utterance pairs with difficulty gradings that are representative of usage scenarios of deployed applications
2. Be compared against dataset variations with randomly generated utterance pairings to ensure robust evaluation outcomes
3. Have an equal number of same speaker and different speaker utterance pairs for each speaker
4. Have at least 500 different speaker utterance pairs for each speaker
5. Have an equal number of utterance pairs per speaker
6. Have an equal proportion of utterance pairs of each difficulty grading per speaker

These criteria cannot overcome the inherent shortcomings of the VoxCeleb 1 dataset, but they control the representativeness and sufficiency of speakers and utterances across groups to reduce variability in evaluation outcomes and improve the robustness of the evaluation across speaker groups. In this way, they mitigate bias in the evaluation process.

Considering Criteria as Pattern Instantiations

While these criteria are specific to the speaker verification domain, they represent specific instances of generalisable ideas. One way of capturing the ideas in a more general form is to consider the criteria as speaker verification specific instantiations of patterns in the *Benchmark Dataset* recipe. Criteria 1. and 2. concern the representativeness and robustness of evaluation sets, and their ability to capture the application context. This makes them instantiations of the *Evaluation datasets* pattern. Criteria 3. concerns the labels of speaker pairs (same or different speaker) and can thus be viewed as an instantion of the *Ground-truth label distribution across groups* pattern. Criteria 4. and 5. concern the representation of groups in the data, and are instantiations of the *Data quantity distribution across groups* patterns. The idea captured by criteria 6. is currently not represented in the pattern catalogue. This leads to the opportunity to capture it as a new pattern.

New pattern: Data quality distribution across groups

Ensure that all groups have sufficient data samples of equivalent difficulty.

This new pattern captures a key insight gained through an empirical study of speaker verification evaluation sets, that found evaluation outcomes to be highly susceptible to the difficulty of data samples used in the evaluation. The pattern has thus been found useful for mitigating bias in a practical context. Like the *Data quantity distribution across groups* pattern, this is a dataset pattern used for analysis purposes that will be most useful in the data gathering stage of the ML development workflow. The pattern has been formulated as a *name/key idea* pair, is unique, and formulated as simply as possible without loosing contextual nuance. It will be included in the *Benchmark Dataset* recipe as a required and unordered pattern. The pattern is meaningful on its own and consistent with other patterns. It satisfies the requirements set out in Chapter 4, though requirements 5, 9 and 11 need to be validated in future work.

5.5. Mitigating Bias Measurement Pitfalls

The previous section investigated how the *Benchmark Dataset* recipe can be used and extended to mitigate shortcomings in speaker verification evaluation sets that can lead to bias. This section compares different approaches to measuring bias in speaker verification to investigate how fairness assumptions, base metrics, bias measures and meta-measures influence the outcome of a bias evaluation. In doing this, the section demonstrates how the *Bias Measurement* recipe can clarify and guide a bias evaluation to ensure that the quantification and measurement of bias leads to reproducible and verifiable claims, and ultimately to systems that work reliably for all users.

The section starts with an overview of the bias and meta-measures that are compared. This is followed by a bias evaluation that empirically compares different measurement approaches, and analyses how base metrics, group-based bias measures and

meta-measures impact the outcomes of a speaker verification bias evaluation. The section concludes with a discussion that contextualises the consequences of biased speaker verification systems in Edge AI applications.

5.5.1. Group-based Bias Measures and Meta-measures

Bias measures that compare error rates across groups typically calculate either differences between, or ratios of error rates. They can be used with any base metric. Below I define the *group-to-min difference* and *group-to-average ratio* bias measures, which can then be used to calculate the *Fairness Discrepancy Rate* and *Normalised Reliability Bias* meta-measures.

Bias measures

Group-to-min difference

This difference-based measure calculates the distance between the base metric (b) of a speaker group (g) and the base metric of the group with the minimum error rate (m). A bias measure of this kind has been used in [85, 135, 263]. This measure is also used to calculate the Fairness Discrepancy Rate (FDR) meta-measure.

$$group\text{-}to\text{-}min\ difference(b_g)_g = b_g - b_m \tag{5.3}$$

Group-to-average ratio

This ratio-based measure calculates the ratio between a group's base metric and the average base metric across all groups. I used this measure in prior work [128] and it is also used to calculate the Normalised Reliability Bias (NRB) meta-measure.

$$group\text{-}to\text{-}average\ ratio(b_g)_g = \frac{b_g}{b_{average}} \tag{5.4}$$

A variation of the vanilla group-to-average ratio is the group-to-average *log* ratio.

$$group\text{-}to\text{-}average\ log\ ratio(b_g)_g = -ln\left(\frac{b_g}{b_{average}}\right) \tag{5.5}$$

Taking the natural logarithm of a performance ratio has intuitive appeal. The bias measure is 0 when the performance of a group equals average performance, negative when the performance is worse than average and positive when it is better than average for the group. The magnitude of the measure is equal for a performance ratio and it's inverse, as $ln(x) = -ln\left(\frac{1}{x}\right)$. This makes the measure interpretable, as bias will be equal in magnitude but opposite in sign for groups that perform half as good (i.e. worse) and twice as good (i.e. better) as average.

Ratio-based measures offer a convenient shortcut for interpretation. The *group-to-average ratio* and *group-to-average log ratio* can be used to approximate relative performance differences as percentage points when taking values of 1 and 0 as 'no difference' respectively.

Meta-measures

Fairness Discrepancy Rate (FDR)

This meta-measure was first proposed by De Freitas Pereira and Marcel [54] to assess fairness in biometric verification systems. The measure performs a pairwise comparison of the false positive rate (FPR) and false negative rate (FNR) differences across all demographic groups (G) at a threshold τ. For each error rate it selects the pair with the maximum difference (i.e. it selects the maximum value of the group-to-min difference measure for g *in* G). The maximum differences are weighted and combined into a joint measure, the FDR. This meta-measure has also been used in [77, 229].

$$\begin{aligned} max_{\Delta FPR}(\tau) &= max\left(\{group\text{-}to\text{-}min\ difference(FPR(\tau))_g\}\right); g\ in\ G \\ max_{\Delta FNR}(\tau) &= max\left(\{group\text{-}to\text{-}min\ difference(FNR(\tau))_g\}\right); g\ in\ G \\ FDR(\tau) &= 1 - (\alpha \times max_{\Delta FPR}(\tau) + (1-\alpha) \times max_{\Delta FNR}(\tau)); 0 <= \alpha <= 1 \end{aligned} \tag{5.6}$$

The FDR ranges from 0 (maximum discrepancy between groups, i.e. most biased) to 1 (minimum discrepancy between groups, i.e. least biased). The measure can be evaluated at different thresholds τ which produce different design error rates $FPR_{average}$, and for different weights α. When $\alpha = 0$ the FDR only considers false negative errors. When $\alpha = 1$, only false positive errors are evaluated.

Normalised Reliability Bias (NRB)

The reliability bias meta-measure has been proposed by Hutiri et al. [131] to measure quality-of-service harms in on-device keyword spotting. The measure calculates the sum of the absolute values of the group-to-average log ratio across all groups in G. To make the reliability bias measure comparable across different group designs, it is normalised by dividing it by the count of groups in G.

$$NRB(b) = \frac{1}{\#G} \sum_{g}^{G} |group\text{-}to\text{-}average\ log\ ratio(b_g)_g| \tag{5.7}$$

The NRB has a lower bound of 0 when the performance across all groups is equal and the model is unbiased. The upper limit is infinite. The higher the score, the more group performance differs from average performance, and the more biased the model. Note that this interpretation of the score is opposite to that of the FDR.

5.5.2. Empirical Comparison

Next, the *Bias Measurement* recipe is used to setup a bias evaluation in which the metrics and measures can be compared. Three optional patterns are excluded from the recipe, as they are not needed to achieve the objectives of this study. Like the experimental setup described in Section 5.4.3, this study is based on the pre-trained end-to-end *ResNetSE34V2* model introduced in Section 5.3.1. The model is evaluated on two evaluation sets constructed from trial pairs in VoxCeleb 1: VoxCeleb 1-H and VoxCeleb 1-Inclusive. VoxCeleb 1-Inclusive has been proposed in Section 5.4 to ensure sufficiency and representativeness across groups, and thus mitigate bias in evaluation sets.

Bias Measurement Recipe

- o 1 ~~Types of harm~~
- o 2 ~~Fairness assumptions~~
- + 3 Group design
- o 4 ~~Metadata approximation~~
- + 5 Disaggregated base metrics
- o 6 Group-based bias measure
  - o Visualising metrics and measures
  - o Meta-measure

5

Group Design

Capture the rationale for choosing attributes that define groups and for deciding when enough different groups have been evaluated.

The pattern is used as is, and includes the following sections:

1. Group attributes
2. Rationale for group design

---

1. Group attributes

Groups based on binary *gender* (male, female) and the intersection of *gender + nationality* have been considered.

2. Rationale for group design

These choices were made due to available metadata released with VoxCeleb 1. The rules used to generate the VoxCeleb1-H and VoxCeleb1-Inclusive evaluation sets precluded the creation of a German male and Irish female speaker group respectively, as insufficient data was available for these groups.

Disaggregated Base Metrics

Calculate performance metrics across groups.

The pattern is used as is, and includes the following sections:

1. Overview of metrics
2. Model evaluation across groups

---

1. Overview of metrics

The model is evaluated with the equal error rate (EER) and minimum detection cost ($min\ C_{Det}$) base metrics. Table 5.10 shows the average results for the performance of the speaker verification model. The average EER and $min\ C_{Det}$ are 50% greater (i.e. the model performs worse) when evaluated on the more challenging conditions of the VoxCeleb1-Inclusive set. Table 5.11 disaggregates the results by speaker group. The $min\ C_{Det}$ base metric for groups was calculated at the threshold which produces the average $min\ C_{Det}$ value in Table 5.10.

Base metric	VoxCeleb1-H	VoxCeleb1-Inclusive
EER	2.402	3.657
min C_{Det}	0.008	0.012

Table 5.10: **Average** speaker verification system performance for equal error rate (EER) and minimum detection cost ($min\ C_{Det}$) base metrics.

2. Model evaluation across groups

For both datasets and base metrics the model performs better for male speakers than for female speakers (see the columns named 'All' for results of the *gender* speaker group). When considering *gender + nationality* groups, the model performs best for US males, with the exception of the $min\ C_{Det}$ on VoxCeleb1-Inclusive, which is lowest for German males. On VoxCeleb1-H, Irish females have the lowest EER and $min\ C_{Det}$, and also perform better than the male speaker groups. On VoxCeleb1-Inclusive Australian females perform best on both base metrics. Their EER is lower than that of US males, the best performing male speaker group. However, their $min\ C_{Det}$ is higher than that of the best performing male group. For male and female *gender + nationality* groups there are speaker groups with significantly worse performance. For example, Norwegian males have an EER that is 304% greater than that of US males when evaluated on VoxCeleb1-H, and 174% greater when evaluate on VoxCeleb1-Inclusive.

Male										
Eval. Dataset	**Base metric**	**All**	**Ireland**	**India**	**US**	**Australia**	**Canada**	**UK**	**Norway**	**Germany**
VoxCeleb1-H	**EER**	**2.289**	2.477	2.230	**1.879**	2.879	2.485	2.215	7.595	-
	min C_{Det}	**0.007**	0.008	0.009	**0.006**	0.007	**0.006**	0.007	0.020	-
VoxCeleb1-Inclusive	**EER**	**3.581**	3.447	3.218	**2.999**	4.362	3.521	3.929	8.210	3.013
	min C_{Det}	**0.011**	0.013	0.018	0.010	0.012	0.011	0.012	0.025	**0.009**
Female										
Eval. Dataset	**Base metric**	**All**	**Ireland**	**India**	**US**	**Australia**	**Canada**	**UK**	**Norway**	**Germany**
VoxCeleb1-H	**EER**	2.564	**1.533**	5.626	2.008	2.524	3.671	2.584	4.880	6.847
	min C_{Det}	0.009	**0.006**	0.020	0.007	0.009	0.011	0.011	0.011	0.010
VoxCeleb1-Inclusive	**EER**	3.757	-	7.028	3.250	**2.788**	4.385	3.969	4.588	10.641
	min C_{Det}	0.012	-	0.023	**0.011**	**0.011**	0.014	0.016	0.014	0.019

Table 5.11: **Disaggregated** speaker verification system performance for equal error rate (EER) and minimum detection cost ($min\ C_{Det}$) base metrics.

Similarly, the EER for German females is 347% greater than that of Irish females evaluated on VoxCeleb1-H, and 282% greater than that of Australian females evaluated on VoxCeleb1-Inclusive. From these results it is clear that the performance of the model varies significantly for speakers of different genders and nationalities.

5.5.3. Impact of Bias Measures on Evaluation Outcomes

The disaggregated base metrics highlight that performance differences exist across groups. However, the extent and impact of the differences are not easy to compare. The *Group-based bias measure* pattern can provide insights on this. However, the outcomes of the comparison may be sensitive to the bias measure that is used. Next, I critically examine how different instantiations of the *Group-based bias measure* pattern impact the outcomes of the bias evaluation.

Group-based Bias Measure

Compare base metrics across groups or between groups and overall performance to assess whether different groups are treated equally.

The pattern is used as is, and includes the following sections:

1. Overview of bias measures
2. Measuring bias across groups
3. Impact and interpretability of bias measure on *Gender* groups
4. Impact and interpretability of bias measure on *Gender + Nationality* groups

---

1. Overview of bias measures

See Section 5.5.1.

2. Measuring bias across groups

Values for the three bias measures were calculated across *gender* and *gender + nationality* speaker groups for the *EER* and $min\ C_{Det}$ base metrics. Table 5.12 shows the measures obtained on VoxCeleb1-Inclusive. When calculating the *group-to-min difference* for the *gender + nationality* speaker group, b_m was selected as the lowest base metric, irrespective of whether it came from a male or female group. Similarly, $b_{average}$ was calculated from all male and female groups for the group-to-average ratios.

3. Impact and interpretability of bias measure on *Gender* groups

Across all bias measures and base metrics the model shows preference for the male group and is prejudiced against the female group. While the bias measures do not alter the evaluation outcomes for the binary group design, they offer different degrees of interpretability. The difference-based *group-to-min difference* is difficult to interpret. As the male group has the smaller error rates and is thus used as reference, the measure evaluates to 0 for males and provides no meaningful insights on this group. For

Male										
Bias measure	**Base metric**	**All**	**Ireland**	**India**	**USA**	**Australia**	**Canada**	**UK**	**Norway**	**Germany**
Group-to-min Difference	**EER**	0.000	0.658	0.429	0.211	1.573	0.733	1.141	5.422	0.224
	min C_{Det}	0.000	0.004	0.010	0.001	0.003	0.002	0.004	0.016	0.000
Group-to-average Ratio	**EER**	0.979	0.942	0.880	0.820	1.193	0.963	1.074	2.245	0.824
	min C_{Det}	0.954	1.088	1.571	0.863	1.046	0.898	1.058	2.120	0.749
Group-to-average log Ratio	**EER**	0.021	0.059	0.128	0.198	-0.176	0.038	-0.072	-0.809	0.194
	min C_{Det}	0.047	-0.084	-0.452	0.148	-0.045	0.108	-0.057	-0.751	0.289
Female										
Bias measure	**Base metric**	**All**	**Ireland**	**India**	**USA**	**Australia**	**Canada**	**UK**	**Norway**	**Germany**
Group-to-min Difference	**EER**	0.176	-	4.240	0.462	0.000	1.596	1.180	1.799	7.853
	min C_{Det}	0.001	-	0.015	0.002	0.002	0.005	0.008	0.005	0.011
Group-to-average Ratio	**EER**	1.027	-	1.922	0.889	0.762	1.199	1.085	1.254	2.909
	min C_{Det}	1.059	-	1.986	0.937	0.945	1.203	1.395	1.208	1.662
Group-to-average log Ratio	**EER**	-0.027	-	-0.653	0.118	0.271	-0.181	-0.082	-0.227	-1.068
	min C_{Det}	-0.057	-	-0.686	0.065	0.056	-0.185	-0.333	-0.189	-0.508

Table 5.12: Three bias measures evaluated for *gender* and *gender + nationality* speaker groups on the VoxCeleb-Inclusive evaluation set.

the female group the values produced by the bias measure are two orders of magnitude apart between the EER and $min\ C_{Det}$ base metrics. This makes it impossible to compare the output values of the bias measure across the two base metrics. The two ratio-based measures can be more readily interpreted. For the male group the *group-to-average ratios* are less than 1 and the *group-to-average log ratios* are positive, implying that this group performs better than average, regardless of the bias measure and base metric used. For the female group the opposite is true.

The values of the two ratio-based bias measures can be interpreted as percentage points that imply similar degrees of bias for the female group. The female EER is ~2.7% worse than average and the $min\ C_{Det}$ ~5.9% or ~5.7% worse than average, depending on whether the vanilla ratio or the log ratio is used. When considering the magnitude of bias for the male group, the merit of the log ratio becomes evident. The male EER is ~2.1% better than average and the $min\ C_{Det}$ ~4.6% (vanilla ratio) or ~4.7% (log ratio) better than average. However, the log ratio, which is centered around 0, is easier to interpret than the vanilla ratio, which requires mental arithmetic to subtract the bias value from 1.

4. Impact and interpretability of bias measure on *Gender + Nationality* groups

For the *gender + nationality* groups the base metric that is used to calculate the bias measure impacts the outcomes of the bias evaluation. When bias measures are calculated with the EER base metric, the order of speaker group performance is maintained. For example, Australian females have the lowest EER and are used as reference for the *group-to-min difference*, evaluating to 0. This group also has the lowest *group-to-average ratio* of 0.762 and the highest *group-to-average log ratio* of 0.271. All bias measures thus show that this speaker group is strongly preferred when considering

the EER base metric.

However, these observations no longer hold true when considering bias based on the $min\ C_{Det}$ base metric. Now German males have the lowest $min\ C_{Det}$ value and are used as reference for the *group-to-min difference*, evaluating to 0. Canadian males, US female and US males have *group-to-min difference* values that are equal to or lower than that of Australian females. When analysing the *group-to-average ratio*, German males are still the most favoured speaker group, followed by US males, Candian males, US females and only then Australian females. While the magnitude of the bias values is not maintained, this order of preference is retained when considering the *group-to-average log ratio*.

5.5.4. Impact of Meta-measures on Evaluation Outcomes

A meta-measures calculates a single value that is used to determine the extent of bias of a system. The *Meta-measure* pattern is thus useful for comparing different models, and for considering bias during model selection. However, just as the base metrics and bias measures have been shown to impact the outcomes of a bias evaluation, the meta-measure may also impact it. Next, I analyse how two instantiations of the *Meta-measures* pattern, the Fairness Discrepancy Rate (FDR) and Normalised Reliability Bias (NRB), impact bias evaluations and their conclusions.

Visualising metrics and measures

Plot metrics and measures across groups to visually examine performance disparities.

The pattern is used together with the meta-measure pattern below.

Meta-measure

Aggregate bias measures across groups into a single bias measure for the model to compare it against other models.

The pattern is used as is, and includes the following sections:

1. Overview of meta-measures
2. Bias evaluation with the Fairness Discrepancy Rate
3. Bias evaluation with Normalised Reliability Bias

---

1. Overview of meta-measures

Meta-measures can be computed for different base metrics. An alternative to the previously introduced EER and $min\ C_{Det}$ is to calibrate speaker verification systems to attain a required average false positive rate (FPR_{avg}). This can be done by choosing a threshold that results in the corresponding FPR_{avg} value. Once the threshold is set, it

also determines an associated average false positive rate (FNR_{avg}). This analysis selected thresholds that calibrate the system to FPR_{avg} of $\{0.001, 0.01, 0.025, 0.05, 0.1\}$. As with the other base metrics, the FPRs and FNRs of speaker groups will deviate from those of the average system, unless the group performance equals average performance.

For each FPR_{avg}, the FDR was evaluated at α ={0, 0.25, 0.5, 0.75, 1}. Recall that $\alpha = 0$ only considers bias due to the *FNR* base metric, while $\alpha = 1$ only considers bias due to the *FPR* base metric. The NRB, on the other hand, was evaluated for the EER and $min\, C_{Det}$ base metrics, the FPR at the FPR_{avg} design error rates and the corresponding FNR, also at the design error rates. These base metrics have been chosen to enable a comparison between the FDR and NRB.

2. Bias evaluation with the Fairness Discrepancy Rate

Figure 5.11 visualises the results of the bias evaluation using the FDR meta-measure for *gender* and *gender + nationality* groups. For all thresholds and $alpha$ the FDR is very close to 1 when considering *gender* groups. This implies that there is almost no difference in performance for males and females. A closer inspection of $FPR_{avg} = 0.001$ reveals that increasing α from 0 (light blue) to 1 (dark green), which effectively increases the weight of the FPR base metric, also increases the FDR value. This means that weighing the FPR reduces bias, or conversely that the FNR has a greater impact on bias at this threshold. At $FPR_{avg} = 0.1$ the opposite is the case: increasing α from 0 (light blue) to 1 (dark green) reduces the FDR, which implies that the FPR increases bias. This effect is exacerbated for the *gender + nationality* groups, where it is clear that for systems calibrated to smaller FPR_{avg}, e.g. 0.001, smaller α (i.e. the FNR) increase bias. This trend reverses for systems calibrated to larger FPR_{avg}, e.g. 0.1, where larger α lower the FDR, implying that the FPR increases bias.

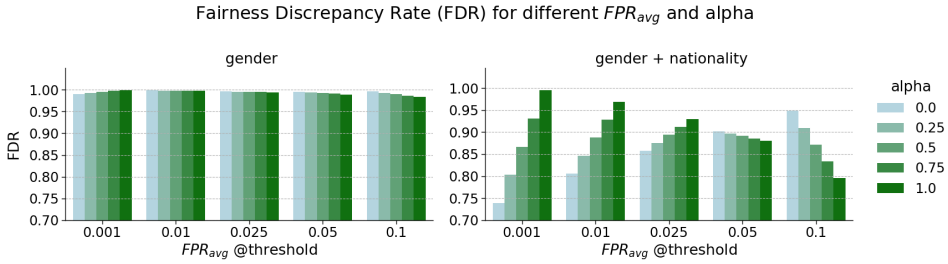


Figure 5.11: FDR meta-measure evaluated for *gender* and *gender + nationality* speaker groups on the VoxCeleb 1-Inclusive evaluation set. The meta-measure is calculated for different values of α and for systems calibrated to thresholds a required average FPR performance.

3. Bias evaluation with Normalised Reliability Bias

Figure 5.12 visualises the results of evaluating bias with the NRB meta-measure for *gender* and *gender + nationality* groups. The analysis compares the EER (purple) and

$min\ C_{Det}$ (orange) base metrics, as well as the FPR (green) and FNR (blue-grey) at systems calibrated to FPR_{avg} values of $\{0.001, 0.01, 0.025, 0.05, 0.1\}$. The NRB, like the FDR, shows that *gender* groups are subjected to less bias than *gender + nationality* groups. For both group designs NRB calculated with the FPR base metric increases as FPR_{avg} decreases. For the FNR base metric the opposite effect is true: the NRB is larger for the FNR base metric at larger FPR_{avg}. To illustrate, at $FPR_{avg} = 0.1$ (i.e. left side of the chart), the NRB calculated with the FPR is lower than the NRB calculated with the FNR. However, at $FPR_{avg} = 0.001$ (i.e. right side of the chart), the NRB calculated with the FPR is greater than the NRB calculated with the FNR. A higher NRB value implies that the system is more biased. This bias evaluation thus leads to the following conclusion. When the system is calibrated to a smaller FPR_{avg}, e.g. 0.001, the FPR increases bias. Conversely, when the system is calibrated to a larger FPR_{avg}, e.g. 0.1, the FNR increases bias. The NRB meta-measure thus draws the opposite conclusion of what the FDR suggests.

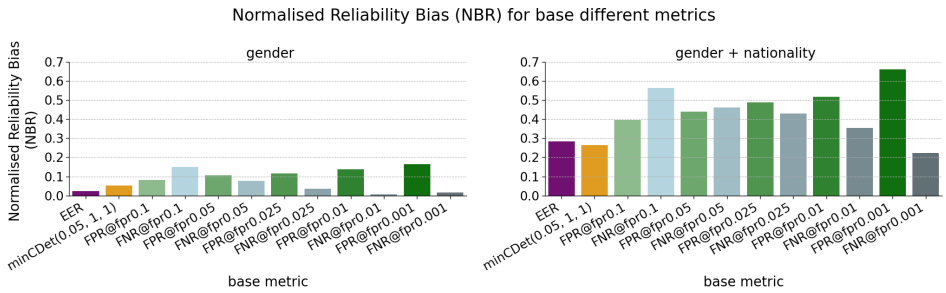


Figure 5.12: NRB meta-measure evaluated for *gender* and *gender + nationality* speaker groups on VoxCeleb 1-Inclusive. The meta-measure is calculated for different base metrics.

5.5.5. Considerations for Bias Measurement

Ratio-based bias measures offer better interpretability

Overall, when compared against each other, the three bias measures that have been evaluated preserve the performance ranking of groups for the same base metric. In particular, the *group-to-average ratio* and *group-to-average log ratio* lead to similar conclusions about the direction and extent of bias. However, the log ratio makes it easier to interpret bias values with a single glance. While the ratio-based measures offer comparative evaluations even for small metric values, the *group-to-min difference* looses its sensitivity when the base metrics are small. Moreover, the extend of bias cannot be compared across base metrics with different orders of magnitude when using the *group-to-min difference*. All bias measures are strongly influenced by the base metric. The order of preference and extent of bias of speaker groups can change for different base metrics.

Meta-measures can be contradictory

The bias evaluations produced with the FDR and NRB meta-measures lead to contradictory conclusions. To illustrate this contradiction, consider a hypothetical speaker verification system that will be used in an application where security is of importance. This necessitates a low FPR, and the system is thus calibrated to $FPR_{avg} = 0.001$. Given the importance of the FPR, bias is then evaluated specifically for this base metric. The NRB of this system for *gender + nationality* groups can be estimated from Figure 5.12 as ∼0.65, which indicates that substantial performance discrepancies exist across speakers of different genders and nationalities. In fact, this base metric shows the highest bias value for the NRB. Based on this bias evaluation, the system should not be used as is, as the security of some groups of users will be severely jeopardised. In a second bias evaluation the FDR is computed with $\alpha = 1$, which weights the meta-measure to only consider bias due to the FPR. From Figure 5.11 a bias value of ∼0.99 is obtained for *gender + nationality* groups. This value suggests that the model contains minimal bias and is safe to use. Indeed, this base metric shows the least bias for the FDR meta-measure.

How can these two meta-measures lead to diametrically opposed conclusions? To investigate this, Table 5.13 shows the FPR base metric, and the *group-to-min difference* and *group-to-average log ratio* bias measures at a design $FPR_{avg} = 0.001$. The results are disaggregated across *gender + nationality* speaker groups. Given the small FPR values, the *group-to-min differences*, which contribute to the calculation of the FDR, are equally small. Using Equation 5.6 with $\alpha = 1$, the FDR for this system is then:

$$\begin{aligned} FDR(@FPR_{avg} = 0.001) &= 1 - (1 \times 0.0053) + 0 \\ &= 0.9947 \end{aligned}$$

As the above calculation shows, the FDR is primarily influenced by the order of magnitude of the base metric, and secondly by the value of $alpha$. When the order of magnitude of the base metric is small, this meta-measure is thus prone to underestimate bias. On the other hand, the *group-to-average log ratio* which is used to calculate the NRB captures a relative relationship and is unaffected by the order of magnitude of the base metric. Using Equation 5.7, the NRB is:

$$\begin{aligned} NRB(FPR_{avg} = 0.001) = \frac{1}{15} \times (&0.3016 + 1.6594 + 0.9124 + 0.6339 + 0.4732 \\ &+ 1.3079 + 0.6539 + 1.2005 + 0.4750 + 0.4716 \\ &+ 0.08 + 1.1872 + 0.3175 + 0.2485) \\ = 0.6615 \end{aligned}$$

Discussion of consequences

While the NRB leads to the conclusion that the system is biased, it is worth asking if the seemingly small differences in FPR actually matter. To explore this, consider another hypothetical scenario. Imagine that an attacker gains access to the device with

Male								
Metric / Measure	**Ireland**	**India**	**USA**	**Australia**	**Canada**	**UK**	**Norway**	**Germany**
FPR@fpr0.001	0.0007	**0.0053**	0.0004	0.001	0.0005	**0.0016**	**0.0037**	**0.0019**
Group-to-min Diff	0.0003	0.0049	0.0000	0.0006	0.0001	0.0012	0.0033	0.0015
Group-to-avg log Ratio	-0.3016	1.6594	-0.9124	0.0000	-0.6339	0.4732	1.3079	0.6539
Female								
FPR@fpr0.001	-	**0.0033**	0.0006	**0.0016**	0.0009	**0.0033**	**0.0014**	**0.0013**
Group-to-min Diff	-	0.0029	0.0002	0.0012	0.0005	0.0029	0.0010	0.0009
Group-to-avg log Ratio	-	1.2005	-0.4750	0.4716	-0.0800	1.1872	0.3175	0.2485

Table 5.13: FPR, group-to-min difference and group-to-average log ratio at design $FPR_{avg} = 0.001$ disaggregated across *gender + nationality* speaker groups, evaluated on VoxCeleb 1-Inclusive. Error rates in **bold** are greater than the design FPR_{avg}.

the previously described speaker verification system. They attempt to access sensitive information on the device by invoking the system once a minute, which is 60 times per hour. At a FPR of 0.001 they have a 1 in a 1000 chance of gaining access to the system. After attempting to access the system for 17 hours, they are likely to have achieved success. However, at a FPR of 0.0053 they now stand a 1 in 190 chance of accessing the system. This means that they only need to attempt to invoke it for 3 hours before they achieve success. This increased exposure to successful attacks presents greater risk of harm for groups that are subjected to worse than average performance. The NRB thus correctly identifies this system as biased, while the FDR misrepresents the potential risk.

This analysis emphasises two important considerations for bias evaluations. Firstly, using the *Group-based bias measure* and *Meta measure* patterns without carefully considering the application context can lead to contradictory claims and erroneous conclusions. Secondly, even though the *Types of harm* and *Fairness assumptions* patterns are optional patterns in the *Bias Measurement* recipe, they contextualise bias evaluations and provide the necessary background to choose between base metrics, bias measures and meta-measures.

5.6. Conclusion

This chapter studied the utility of the identified patterns for bias detection and mitigation in a machine learning (ML) use case to validate them against requirements 6, 7 and 8 listed in Chapter 4. The use case focused on speaker verification systems, which are important for securing voice-activated Edge AI. Prior to the research underlying this chapter, bias in speaker verification had only been addressed in a very limited number of studies. This made speaker verification a suitable use case for validating the patterns in a new domain. The empirical and analytical studies in Section 5.3, and in Sections 5.4 and 5.5 demonstrate that the patterns in the *Benchmark Dataset* and

Bias Measurement recipes, and the processes that the recipes prescribe, support bias detection and mitigation in practice. Moreover, the studies show that the recipes and patterns enable a coherent, reproducible and consistent bias evaluation and that they help to interrogate the impacts of design choices. They thus satisfy the functional requirements that this chapter set out to validate.

The patterns and recipes were validated and improved in two design iterations. In the first develop-demonstrate-validate iteration in Section 5.3 the *Benchmark Dataset* recipe was used in retrospective mode to detect sources of bias in the benchmark evaluation sets of a popular speaker verification competition. The *Bias Measurement* recipe was used in proactive mode to set up a bias evaluation and detect sources of bias in baseline models released for the speaker verification competition. When using the patterns in practices, it was found helpful to divide the discussion of each pattern into sections. For example, the *Types of harm* pattern includes three sections that discuss types of errors, a taxonomy of risks and the consequences that these have on bias. The *Metadata approximation* pattern discussed the approximation approach, the consequences this has for evaluation and critically reflected on the metadata approximation. While the section titles are mostly specific to individual patterns, they are general and can be reused in future bias evaluations. I thus view them as emergent, more detailed pattern templates. This iteration also resulted in the addition of a new dataset pattern to ensure that the data source is legitimate and that the dataset contains no unforeseen bias. This *Data source* pattern was incorporated in the *Benchmark Dataset* recipe as a required, unordered pattern.

The second develop-demonstrate-validate iteration examined evaluation sets and bias measurement more closely. Section 5.4 conducted an empirical study to examine how the quantity and difficulty of utterance pairings across groups in evaluation sets affect evaluation outcomes. Based on the results, criteria were proposed for creating speaker verification evaluation sets. The criteria can be considered as speaker verification specific instantiations of patterns in the *Benchmark Dataset* recipe. This framing led to the discovery of a new pattern, *Data quality distribution across groups*, to ensure that all groups have sufficient data samples of equivalent difficulty. The pattern was incorporated in the *Benchmark Dataset* recipe as a required, unordered pattern. In Section 5.5 the *Bias Measurement* recipe was used to examine and mitigate measurement pitfalls during bias evaluations. This section showed empirically that measurement choices during a bias evaluation impact the outcomes of the evaluation. The patterns in the *Bias Measurement* recipe helped to make these choices explicit and to compare them, thus clarifying and guiding the bias evaluation process.

Together, these two design iterations and the careful analysis of the patterns in practice demonstrate that the patterns elicited in the previous chapter are useful for detecting and mitigating bias in a ML use case. This chapter thus successfully addresses Research Question 2 and accomplishes Goal 2. The next chapter will extend the patterns to the Edge AI domain, and investigate their utility for detecting and mitigating

bias in an on-device keyword spotting use case. This use case will validate the remaining requirements 9 and 11, and will address the outstanding research goal and question.

6

Extending Patterns to Edge AI

This chapter extends and adapts the patterns to on-device machine learning (ML) in a keyword spotting use case, a prominent application of Edge AI. Prior to the research supporting this chapter, no studies had been done on bias in on-device ML. The goal of this research was thus to use the patterns to investigate sources of bias in on-device ML, and to propose mitigating actions. In contrast to the previous chapter where the patterns and recipes were used to analyse pre-trained models, this chapter embeds the recipes in the model development process to uncover potentially new and unknown sources of bias. The empirical study presented in the chapter shows that the patterns in the Benchmark Dataset and Bias Mitigation recipes support bias detection and mitigation during Edge AI model development. This validates that the patterns are reusable and adaptable to different contexts. The insights derived from the bias evaluation are captured as two new patterns concerned with model selection and design choices. These patterns extend the pattern catalogue to bias detection and mitigation in on-device ML.

This chapter is based on the following publications:

1. W. Hutiri, A. Y. Ding, F. Kawsar, and A. Mathur. Tiny, Always-on and Fragile: Bias Propagation through Design Choices in On-device Machine Learning Workflows. *ACM Transactions on Software Engineering and Methodology*, 4 2023. ISSN 1049-331X. doi: 10.1145/3591867 [131]
2. W. Toussaint, A. Mathur, A. Y. Ding, and F. Kawsar. Characterising the Role of Pre-Processing Parameters in Audio-based Embedded Machine Learning. In *The 3rd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT '21)*, pages 439–445, Coimbra, Portugal, 2021. ACM. ISBN 9781450390972. doi: 10.1145/3485730.3493448 [292]

6.1. Introduction

This chapter presents the final iteration of the design science research cycle. The chapter extends and adapts the patterns that have been validated in a speaker verification use case in Chapter 5 to the on-device machine learning (ML) domain. Where the previous chapter used patterns and recipes to analyse pre-trained models, this chapter reuses the *Benchmark Dataset* and *Bias Mitigation* recipes proactively to detect sources of bias during model development in an on-device keyword spotting use case. The chapter thus extends and adapts the patterns and recipes to a different context that demonstrates their utility for Edge AI applications, while also validating requirements 9 and 11.

Keyword spotting (KWS) systems are a prominent application area of on-device ML. As illustrated in Section 2.5, they are an integral component of voice-activated Edge AI, where they enable interactions with digital services on smart speakers and smart phones. Beyond convenience, voice activation holds promise to increase the accessibility of digital services for individuals who have impaired vision, restricted mobility and movement. Oftentimes the applications where KWS can have the greatest positive impact are those that serve vulnerable user groups, for example emergency response and home care to serve sick, elderly and differently-abled people. Many commercial products are targeted at these applications and promise capabilities such as voice-activated urgent response (e.g. "call help") with on-device KWS [234, 298]. People that use these products place confidence in their ability to support them in moments of crisis and provision them with access to critical care services.

Despite the evident societal promise of on-device KWS, human speech signals exhibit variability based on social and physiological attributes of the speaker [108]. Moreover, the on-device ML setting poses hardware constraints that limit available memory, computing capabilities, and energy resources during inference [13]. These factors can lead to bias in on-device KWS systems and make it necessary to detect and mitigate bias during their development.

This chapter uses patterns to investigate bias in the development workflow of on-device KWS systems. It largely draws on prior research published in [131]. The chapter starts with an overview of the on-device KWS use case in Section 6.2, highlighting technical details, design choices and constraints that arise during on-device ML development. Next, Section 6.3 describes an empirical study that investigates design choices in on-device ML development workflows as potential sources of bias in KWS systems. This section uses the patterns and processes captured in the *Benchmark Dataset* and *Bias Mitigation* recipes to support a bias evaluation during model training and optimisation. Section 6.4 presents the study results, first analysing the impact of design choices during model training, and then during model optimisation. Based on the results, Section 6.5 proposes the *Bias-aware Model Selection* and *Data-driven Design Decisions* patterns for mitigating bias in on-device ML. The chapter concludes in Section 6.6.

6.2. On-device Keyword Spotting Use Case

This section briefly introduces keyword spotting (KWS) systems. It then proceeds to map design choices in the on-device ML workflow, first discussing the data processing stages, and then highlighting the various constraints, intervention strategies and design choices that a practitioner encounters while designing on-device ML systems in practice. Finally, the impact of design choices on on-device KWS is considered. For a primer on on-device ML and its relation to Edge AI, I refer the reader to Section 2.4.

6.2.1. Keyword Spotting Overview

An audio KWS system as shown in Figure 6.1 takes a raw speech signal as input and outputs the keyword(s) present in the signal from a predefined set of keywords. The end-to-end training and inference pipeline of a KWS system works as follows. First, a raw speech signal is sampled from the microphone at a predefined sample rate (e.g. 8KHz, 16KHz) and split into overlapping, short time duration frames using a sliding window approach. This framing operation requires that a number of *pre-processing parameters* are specified, which include:

1. a frame length that defines the duration of each frame,
2. a frame step that indicates the step size by which the sliding window is moved,
3. a window function which helps in reducing spectral leakage in Discrete Fourier Transform (DFT)

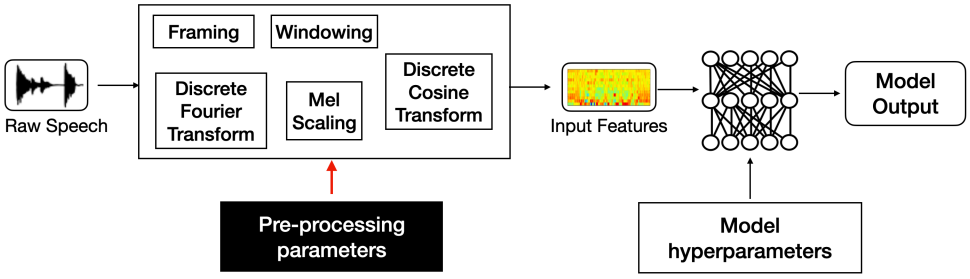


Figure 6.1: Audio processing pipeline during training and inference

Thereafter, each frame of the speech signal is transformed into *input features*: first, a DFT is applied to each frame to obtain log-scaled filter bank features known as log Mel spectrograms. Optionally, log Mel spectrograms can be de-correlated using a Discrete Cosine Transform to generate Mel Frequency Cepstral Coefficiencts (MFCCs). The number of log Mel spectrograms and MFCCs is also a designer-chosen parameter, often tuned empirically. Finally, the frame-level features (log Mel spectrograms or MFCCs) are concatenated across frames and mean-normalized to form a two-dimensional representation of the speech signal which is used to train a deep neural network classifier, as described in [35]. This process also involves choosing an appropriate *neural*

network architecture that satisfies the resource constraints of the deployment device. Optionally, an ML engineer can also choose to optimise the trained neural network by applying various *model compression* techniques such as weight pruning.

6.2.2. Design Choices in On-device ML

Heterogeneous devices, diverse users and unknown usage environments make the performance of on-device ML highly context dependent. During development, practitioners are faced with a large number of decisions to choose interventions that overcome hardware constraints and meet operational demands. Collectively, constraints and context-dependency make on-device ML development a complex engineering undertaking that requires mastery of hardware, software engineering and data processing techniques, alongside an in-depth understanding of the application context.

Data Processing Workflow for On-device ML

The key processing steps during on-device ML development are model training, interventions, and inference. A typical data processing pipeline for on-device ML, as shown in Figure 6.2, consists of familiar ML processing steps for model training, evaluation, selection and inference. Key differences between on-device ML and cloud-based ML development arise due to the low compute, memory and power resources of end devices [60]. To enable on-device inference, interventions are needed to optimise a trained model and its data processing pipeline for on-device deployment. These aspects are described in greater detail below.

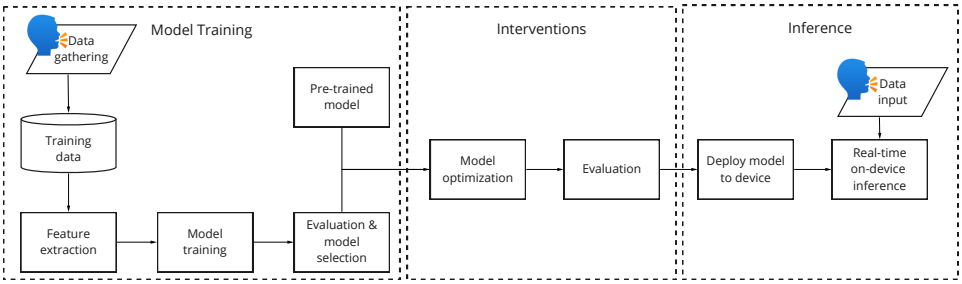


Figure 6.2: Data processing pipeline for on-device machine learning development

Training

The dominant approach for developing on-device ML is to delegate resource-intensive model training to the cloud and to deploy trained and optimised models to devices [60]. The approach for training models is similar to typical ML pipelines: input data is gathered and undergoes a number of pre-processing operations to extract features from it. Thereafter, ML models are trained, evaluated and selected after optimising a loss function on the data. Pre-trained models can also be downloaded and used if training data or training compute resources are not available.

Interventions

The key differences between on-device ML and cloud-based ML development arise due to the low compute, memory and power resources of end devices [60]. To enable on-device deployment of the trained model, various interventions are needed to optimise the model and its data processing pipeline. Common interventions include techniques such as model pruning, model quantisation, or input scaling; all of which are aimed at optimising device-specific performance metrics such as response time or latency [13], memory consumption [107], or energy expenditure [329] with minimal impact on the model's accuracy.

Inference

Once deployed, the trained and optimised model is used to make real-time, on-device predictions. On-device inference performance is determined by the model training process, from data collection to model selection, and the real-time sensor data input, but also by deployment constraints and interventions applied to the model.

Navigating Design Choices in On-device ML Engineering

Based on the on-device data processing workflow in Figure 6.2, an engineer has to make several key design choice. The design choices arise because various interventions need to be taken to overcome constraints of on-device ML. These interventions can impact the accuracy and bias of on-device ML models.

Map of Design Choices

To build on-device ML, engineers need to navigate deployment constraints and interventions alongside ML training and deployment. This is technically challenging, and charges practitioners with the responsibility to take design actions and make design choices at each development step. Importantly, as on-device deployment constraints require interventions in the development process, design choices like the choice of model architecture, sample rate, input features and model compression techniques affect predictive and hardware performance, as well as bias [292]. Even though some design choices can be optimised through automated experimentation, iterating through all possible values requires extensive computing resources and time. This increases the cost of training, which diminishes the usefulness of on-device ML as an accessible, low-cost technology. Moreover, each design choice can introduce bias into the system. If time or compute are limited, engineers may need to limit the extent of their experimentation and only focus on a small set of choices.

Some of the key design choices are visualised as a decision map in Figure 6.3. The availability of training data is a logical starting point for development, as it determines whether a new model can be trained, or if a pre-trained model must be downloaded. Once an engineer commits to the design action of training a new model, they are confronted with design choices to select an algorithm, hyper-parameters, input features, pre-processing parameters and a data sample rate. After training or downloading the

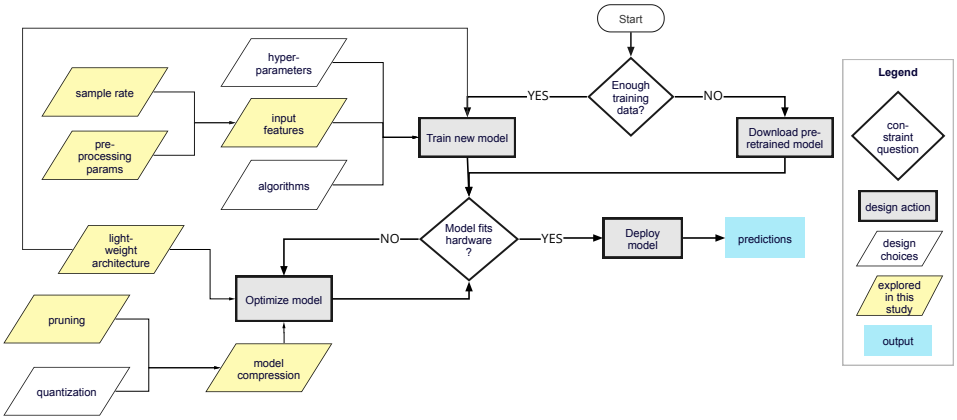


Figure 6.3: Decision map of design choices during on-device ML engineering. Yellow chart elements are design choices studied in this thesis.

model, the engineer needs to determine if it fits within the memory, compute and power budget of the device. If it does, the model can be deployed to make predictions. Else, the engineer must take design actions to optimise the model and reduce its resource requirements. This can be done through interventions like training a more light-weight architecture or compressing the model. These choices present further sub-choices, for example model compression can be done with pruning, quantisation or both. In comparison to quantisation, pruning involves more hyper-parameters and thus requires more design choices. Each design choice modifies the model, and has the potential of introducing bias in its predictions.

6.2.3. Impact of Design Choices on KWS

Prior literature has studied how design choices impact the performance of KWS systems. Next, this literature is reviewed along the lines of design choices as they appear in the on-device ML workflow.

Sample Rate

The sample rate can be seen as a deployment constraint due to hardware limitations such as microphone capabilities [203]. Prior works have also used the sample rate as a tunable parameter to adjust the power consumption on an embedded device during data collection [61, 273]. However, to our knowledge, there has been no study on how the choice of sample rate affects bias of ML models, other than our prior work [292] which is extended by this study.

Pre-processing Parameters

The choice of audio pre-processing parameters is known to have an impact on the performance of a KWS model in an embedded system [203]. Frame length and frame step

together determine the temporal dimension of the 2D features that are fed to a DNN. The number of log Mel spectrogram or MFCC features determine the length of features in each time segment. Together, these features influence the dimensions of the input data to the model, which in turn impacts the number of computations during inference. This insight was used in a recent work named ePerceptive [203], wherein the authors experimented with different values of the frame step to achieve a good trade-off between inference accuracy and latency. However, there has been no prior work which has explored potential accuracy-bias trade-offs due to pre-processing parameters.

Model Architecture

Unsurprisingly, the model architecture plays an important role in the performance of KWS systems. Inference on model architectures with fewer parameters takes less time, but could lead to accuracy degradation. On the contrary, deeper models with a large number of parameters might provide better accuracy, at the expense of higher inference latency. Prior KWS works [7, 35, 112, 120, 296, 334] have experimented with different architectures to achieve a good accuracy-latency trade-off. However these studies have not evaluated bias in KWS systems due to the choice of model architecture.

Pruning Hyper-parameters

Finally, when compressing a trained model for deployment using model pruning, an engineer needs to specify a number of parameters such as the final sparsity, pruning frequency, pruning schedule and learning rate. The final sparsity, specified as a percentage, determines the proportion of weights that will be set to 0 during model pruning. A high final sparsity leads to more compressed models, which result in lower storage requirements and reduced inference latency on the device [171]. The pruning frequency is an integer value that indicates after how many training steps the model should be pruned. The pruning schedule can take two values: i) constant sparsity, which specifies a fixed sparsity level of the model throughout the training, or ii) polynomial decay, where the initial pruning sparsity grows rapidly in the beginning, but then plateaus slowly to the final sparsity. Finally, the pruning learning rate controls the step size taken by the model optimiser (e.g. Adam or stochastic gradient descent) during backpropagation.

Prior literature on neural network pruning primarily investigated the impact of the final sparsity on model accuracy [171] and does not shed light on the impact of other pruning parameters. However, given that these parameters also constitute important design decisions during model optimisation, they are investigated in this study.

6.3. Studying Design Choices as a Source of Bias

Having established that the impact of design choices on bias has not been studied in the KWS or on-device ML literature, we set up empirical experiments to investigate

design choices related to important design actions for on-device ML: model training and model optimisation. This section introduces the empirical study and describes the experiment design, training and evaluation datasets, training details and the bias evaluation setup. The *Benchmark Dataset* recipe has been used to construct KWS datasets for training and evaluation. The *Bias Measurement* recipe was used to guide the bias evaluation.

6.3.1. Experiment Design

Informed by the on-device ML development workflow presented in Figure 6.2 and our prior work [292], the objective of this study is to evaluate the impact of design choices and choice variables on model accuracy and bias in the development workflow of an on-device audio KWS system. The design choices and choice variables that are considered have been summarised in Table 6.1.

Design action	Design choice	Choice variable (unit)	Variable values
Train new model	input features | sample rate	sample rate (kHz)	8, 16
Train new model	input features | pre-processing	feature type	log Mel spectrogram, MFCC
Train new model	input features | pre-processing	# Mel filter banks	20, 26, 32, 40, 60, 80
Train new model	input features | pre-processing	# MFCCs	None, 10, 11, 12, 13, 14
Train new model	input features | pre-processing	frame length (ms)	20, 25, 30, 40
Train new model	input features | pre-processing	frame step (% frame length)	40, 50, 60
Train new model	input features | pre-processing	window function	Hamming, Hann
optimise model	light-weight architecture	model architecture	CNN, low latency CNN [251]
optimise model	model compression | pruning	final sparsity (%)	20, 50, 75, 80, 85, 90
optimise model	model compression | pruning	pruning frequency	10, 100
optimise model	model compression | pruning	pruning schedule	constant sparsity, polynomial decay
optimise model	model compression | pruning	pruning learning rate	1e-3, 1e-4, 1e-5

Table 6.1: Overview of design choice variables and values investigated in this study

Architectural design choices

As discussed in Section 6.2, the neural network architecture is an important design choice during model training. We study two convolutional neural network (CNN) architectures for KWS, originally proposed in [251] and later implemented in the TensorFlow framework. The architecture referred to as *CNN* consists of two convolutional layers followed by one dense hidden layer, while the low-latency CNN (*llCNN*) consists of one convolution layer followed by two dense hidden layers. Sainath and Parada [251] showed that the llCNN architecture, by virtue of having less convolution operations, is more optimised for on-device KWS.

Input feature design choices

The next area of interest are choices that affect the input features of a model, namely the sample rate and pre-processing parameters. Audio KWS developer benchmarks often use a 16kHz audio input [188, 317]. In practice many devices collect data at a

lower sample rate of 8kHz [203] due to hardware constraints. We thus train models with audio data at two sample rates, 16kHz and 8kHz, for both architectures. For studying the impact of pre-processing parameters, we take inspiration from prior KWS literature [7, 35, 112, 120, 296, 334], and experiment with two feature types, log Mel spectrograms and MFCCs. More specifically, we vary the dimensionality of log Mel spectrograms from 20 to 80, and of MFCCs from 10 to 14. We also consider log Mel spectrograms that are used directly as input features, with no MFCCs. Further, we experiment with three temporal pre-processing parameters: frame length (20-40 ms), frame step (40%-60% overlap) and the window type (Hamming/Hann); these values are based on prior on-device KWS works [7, 112, 120, 203, 334].

Model compression design choices

With regards to model optimisation, the study focuses on model compression, in particular parameter choices during post-training pruning. Based on prior literature, the pruning sparsity is varied from 20% to 90% [171, 172]. For the pruning schedule, constant sparsity and polynomial decay are considered, as explained in Section 6.2.3. The three learning rate values were chosen based on prior KWS literature on model training [7, 35, 112, 120, 296, 334]. For pruning frequency, we considered two values: 100 (the default frequency in TensorFlow) and a faster option of 10, wherein the pruning operation takes place after every 10 training steps.

6.3.2. Training and Evaluation Datasets

Next, the training and evaluation datasets are described. The *Benchmark Dataset* recipe has been embedded in the study design to ensure that datasets are representative, appropriate and sufficient. At the time the study was conducted, the *Data quality distribution across groups* pattern had not yet been identified and thus is not considered. While the recipe was initially intended to be used for evaluation benchmarks, this study repurposes it to create training and evaluation datasets for model building. As the training and evaluation datasets are subsets of the same corpora, they are discussed together.

Benchmark Dataset Recipe

+ 1 Evaluation datasets
+ 2 Group design
o 3 Metadata approximation
  + Ground-truth label distribution across groups
  + Data quantity distribution across groups
  + ~~Data quality distribution across groups~~
  + Data source

Training and Evaluation datasets

Evaluate models on multiple, representative datasets that capture the application context.

The pattern is extended to also include training datasets, and includes the following sections:

1. Attributes of datasets

---

1. Attributes of datasets

Models were trained and evaluated on five spoken keywords datasets spanning four languages: English, German, French, and Kinyarwanda. The datasets are described below.

Google Speech Commands

Google Speech Commands (*google_sc*) [317] is an English language dataset consisting of 104 541 spoken keywords from 35 keyword classes such as *Yes, No, One, Two, Three*, recorded by volunteer contributors and released at a 16kHz sample rate. The study uses the same train, validation and test set splits of 85%, 10%, 5% respectively proposed in the original dataset. During training we randomly sampled from the overrepresented male training set to ensure that mini-batches have an equal balance of male and female speakers.

Multilingual Spoken Words Corpus (MSWC)

The Multilingual Spoken Words Corpus (MSWC) [188] is a large dataset of spoken words in 50 languages, originally sampled at 48KHz. Each language partition contains hundreds of hours of audio data with tens of thousands of keyword classes. MSWC has been derived from Mozilla Common Voice[1] by splitting the crowd-sourced, read-speech corpus into individual words. We chose four of the languages with the largest data resources in MSWC to create four KWS datasets in different languages: MSWC English (*mswc_en*), German (*mswc_de*), French (*mswc_fr*) and Kinyarwanda (*mswc_rw*). Each of the MSWC datasets was created with data from its language partition, and a consistent approach to select keywords, balance data across male and female speakers, and split the dataset into train, validation and test splits. We excluded data where the gender metadata field was empty, 'none' or 'other' to enable us to generate gender-balanced datasets.

For each dataset, we selected keywords from the 35 largest keyword classes to create training datasets that are equivalent to Google Speech Commands. Following the keyword selection strategy of the authors of the MSWC dataset, we only selected keywords with more than 3 characters. Additionally, if two words started with the same 3 letters, we only selected the first ocurring word. This resulted in a total of 200 628 keyword utterances for MSWC English, 85 572 keyword utterances for MSWC German, 75 644 keyword utterances for MSWC French and 53 608 keyword utterances for MSWC Kinyarwanda. The dataset sizes vary based on the language representation in the Mozilla Common Voice corpus.

[1] https://commonvoice.mozilla.org/

To create the dataset splits, we followed the protocol described in [188] as closely as possible while enforcing gender-balance. We first created a list of unique keyword-speaker pairs so that train, validation and test sets are separate. Next, we randomly sampled 80% of keyword-speaker pairs for training. We then randomly sample 10% of keyword-speaker pairs for validation, excluding pairs already in the training set and rounding to the nearest integer. Finally, we allocated the remaining keyword-speaker pairs to the test set for evaluation.

Keyword classes

In KWS tasks, keywords are the ground-truth labels. The keywords considered in this study are listed below.

Google Speech Commands:
bed:0, bird:1, cat:2, dog:3, down:4, eight:5, five:6, four:7, go:8, happy:9, house:10, left:11, marvin:12, nine:13, no:14, off:15, on:16, one:17, right:18, seven:19, sheila:20, six:21, learn:22, stop:23, three:24, tree:25, two:26, up:27, wow:28, yes:29, zero:30, backward:31, follow:32, forward:33, visual:34

MSWC English:
about:0, after:1, also:2, been:3, could:4, first:5, from:6, have:7, however:8, just:9, know:10, like:11, many:12, more:13, most:14, only:15, other:16, over:17, people:18, said:19, school:20, some:21, that:22, they:23, this:24, three:25, time:26, used:27, were:28, what:29, when:30, will:31, with:32, would:33, your:34

MSWC German:
aber:0, alle:1, auch:2, dann:3, dass:4, diese:5, doch:6, durch:7, eine:8, gibt:9, haben:10, hauptstadt:11, heute:12, hier:13, immer:14, jetzt:15, kann:16, können:17, mehr:18, muss:19, nach:20, nicht:21, noch:22, oder:23, schon:24, sein:25, sich:26, sind:27, wenn:28, werden:29, wieder:30, wird:31, wurde:32, zwei:33, über:34

MSWC French:
alors:0, aussi:1, avec:2, bien:3, cent:4, cette:5, comme:6, cest:7, dans:8, deux:9, donc:10, elle:11, fait:12, huit:13, mais:14, mille:15, monsieur:16, même:17, nous:18, numéro:19, plus:20, pour:21, quatre:22, saint:23, sept:24, soixante:25, sont:26, tout:27, trois:28, très:29, vingt:30, vous:31, également:32, était:33, être:34

MSWC Kinyarwanda:
abantu:0, ariko:1, avuga:2, bari:3, benshi:4, buryo:5, cyane:6, gihe:7, gukora:8, gusa:9, hari:10, ibyo:11, icyo:12, igihe:13, imana:14, imbere:15, kandi:16, kuba:17, kugira:18, kuko:19, kuri:20, mbere:21, muri:22, ndetse:23, neza:24, ntabwo:25, nyuma:26, perezida:27, rwanda:28, ubwo:29, umuntu:30, umwe:31, yagize:32, yari:33, yavuze:34

Group design

Capture the rationale for choosing attributes that define groups and for deciding when enough different groups have been evaluated.

The pattern is used as is and includes the following sections:

1. Group attributes
2. Rationale for group design
3. Limitations of group design

---

1. Group attributes

We considered groups based on a speaker's binary gender (male, female).

2. Rationale for group design

Human speech signals exhibit variability based on social and physiological attributes of the speaker [108]. A starting point for investigating bias in on-device audio KWS is thus to investigate inference performance for speaker groups with different demographic attributes, like gender.

3. Limitations of group design

Our approach to labelling voice samples with gender was limited to a binary gender classification system and a crowd-sourced labelling campaign (see the *Metadata approximation* pattern below for a discussion on limitations of crowd-sourced labelling). Even though the MSWC gender labels were self-annotated, binary gender representation excludes individuals that do not fit within this classification system from the bias evaluation. Gender is also just one of many demographic attributes that influences the human voice [271]. Subgroups established along other speaker attributes can reveal further dimensions of bias and should be investigated in future work.

Metadata approximation

Approximate metadata labels from other features in the dataset.

The pattern is used as is and includes the following sections:

1. Approximation approach
2. Critical reflection on metadata approximation

---

1. Approximation approach

The gender metadata in Mozilla Common Voice and thus also the MSWC datasets has been provided by data donors and thus corresponds with the self-identified gender of the speaker. No further metadata approximation was done. For the Google Speech Commands dataset every utterance was labeleed as male or female using a crowd-sourced data labelling campaign conducted on Amazon Mechanical Turk.

2. Critical reflection on metadata approximation

Crowd-sourced labelling can introduce misclassifications [252]. For example, the voice of male and female speakers can be higher or lower pitched than what a data worker perceives as normal for that gender, and misclassified accordingly. Moreover, gender identification from speech is also prone to cultural biases. As we were unable to match the cultural identity of speakers in the dataset with those of data workers, we were unable to control for this variance.

Ground-truth label distribution across groups

Ensure that ground-truth labels are balanced across groups.

The pattern is used as is.

We assumed that the labels in the KWS datasets are correct, that keywords are unambiguous and can be known exactly. The protocol for creating the MSWC datasets ensured that labels are balanced for male and female speaker groups. This was done as follows. For each keyword label in the MSWC datasets we counted the utterances per gender. We included all utterances of the gender with fewer utterances/keyword, and randomly sampled the same number of utterances from the gender with more utterances/keyword. We joined the selected data for both genders and all keywords, before splitting the data into train, validation and test sets.

Data quantity distribution across groups

Ensure that all groups are represented sufficiently in the data.

The pattern is used as is and includes the following sections:

1. Data distribution across datasets and gender

---

1. Data distribution across datasets and gender

dataset split	**MSWC English**	**MSWC German**	**MSWC French**	**MSWC Kinyarwanda**
female training	79002 (39%)	34728 (41%)	31127 (41%)	20713 (39%)
male training	79611 (40%)	34329 (40%)	30276 (40%)	21786 (41%)
female validation	10496 (5.2%)	4613 (5.4%)	2790 (3.7%)	3580 (6.7%)
male validation	10238 (5.1%)	3976 (4.6%)	3601 (4.8%)	1801 (3.4%)
female test	10816 (5.4%)	3445 (4%)	3905 (5.2%)	2511 (4.7%)
male test	10465 (5.2%)	4481 (5.2%)	3945 (%)	3217 (6%)
total	200628	85572	75644	53608

Table 6.2: Audio keyword utterance count (and % of total dataset) across dataset splits for MSWC English, MSWC German, MSWC French and MSWC Kinyarwanda datasets.

In Google Speech Commads female speakers constituted 30% of the original training data, 32% of the validation and 29% of the test data. For the four MSWC datasets the count of utterances and proportion of males and females across the dataset splits is shown in Table 6.2. During training we ensured that mini-batches have an equal balance of male and female speakers for all datasets.

Data source

Ensure that the data source is legitimate and that the dataset contains no unforeseen bias.

The pattern is used as is.

The data samples in Google Speech Commands and Mozilla Common Voice were contributed by volunteers, for the purpose of creating datasets to train and evaluate keyword spotting and speech models respectively. The use of the data in this study aligns with the purpose of the original data collections, to which the volunteers consented. There are thus no evident ethical challenges regarding data sourcing. However, it is important to highlight that volunteer contributions can result in a self-selection bias of volunteers - not all people are equally likely to contribute to volunteer crowd-sourcing efforts. This phenomenon may explain the high representation of male speakers in both datasets.

6.3.3. Training Details

Initial model training

Model training was implemented in Tensorflow 2.0 with a Nvidia V100 GPU. For each dataset we iteratively trained models with all combinations of model architectures, sample rates and pre-processing parameters listed in Table 6.1. This resulted in 3456 candidate models per dataset, and a total of 17280 experiments across 5 datasets. We used the TF HParams API[2] for tuning the learning rate during training for each model from the following three options: $\{1e-2, 1e-3, 1e-4\}$. Optimisation was done with the Adam optimiser and a fixed batch size of 128 samples. Each model was trained for 10 epochs, which was chosen based on empirical evidence that the model performance did not improve beyond 10 epochs.

Model pruning

After training, we used model selection criteria that consider accuracy and bias (discussed in detail in Section 6.5.1) to select baseline models for model compression. Table 6.3 lists the number of baseline models selected per dataset for the pruning experiments. For the Google SC and MSWC Kinyarwanda datasets we could not find models that met all our selection criteria across architectures and sample rates, which is why

[2] https://www.tensorflow.org/tensorboard/hyperparameter_tuning_with_hparams

fewer baseline models were selected for these datasets. We then obtained the compressed version of the baseline models under each combination of pruning parameters listed in Table 6.1. As with training, we used 10 epochs for pruning. The pruning experiments resulted in 72 pruned models for each baseline model, and a total of 12168 experiments.

	Google SC	MSWC German	MSWC English	MSWC French	MSWC Kinyarwanda
16kHz CNN	9	9	9	9	6
16kHz llCNN	8	9	9	9	7
8kHz CNN	9	9	9	9	7
8kHz llCNN	9	9	9	9	6

Table 6.3: Number of baseline models pruned per dataset, architecture and sample rate

6.3.4. Bias Evaluation Setup

To examine design choices as sources of bias during model training and optimisation, we used the *Bias Measurement* recipe to guide the setup of the bias evaluations. Here, the pattern instantiations of the recipe are described. The *Disaggregated base metrics*, *Group-based bias measure* and *Meta-measure* patterns are introduced. The evaluation and visualisation with the *Visualising metrics and measures* pattern follows in Section 6.4.

Bias Measurement

- o 1 ~~Types of harm~~
- o 2 Fairness assumptions
- + 3 Group design
- o 4 Metadata approximation
- + 5 Disaggregated base metrics
- o 6 Group-based bias measure
  - o Visualising metrics and measures
  - o Meta-measure

Fairness assumptions

Make assumptions explicit with an upfront statement about that which is believed to make a system fair.

The pattern is used as is and includes the following sections:

1. Fairness considerations
2. Statement of assumptions

---

1. Fairness considerations

When interacting with services that make use of on-device ML, users are justified to expect reliable performance, irrespective of their demographic, social or economic attributes. Bias in on-device ML can lead to systematic device failures due to performance disparities across user groups. Such failures may affect the services that an

on-device ML system provisions, or even the hardware itself. For example, a KWS system with poor predictive performance can require several user attempts to activate the system. This can increase computations, which leads to increased power consumption and faster drainage of a device's battery.

Bias is a particular concern in on-device settings, as it counteracts the promise of ubiquitous, technology-enabled service access; an important value proposition of on-device ML. If bias remains unidentified and is not accounted for, it can be a source of unfairness in on-device ML systems. Unfair on-device ML systems that are deployed at scale can lead to a discriminatory service infrastructure that restricts who has access to services, and how these services can be accessed. User groups that may draw the most benefits from accessible digital technology like voice-activated emergency response or care services, are also most vulnerable to being subjected to bias as they often present a minority population that is not represented in datasets and engineering teams.

2. Statement of assumptions

We consider an on-device KWS model a reliable device component for a group if the group's predictive accuracy equals the model's overall accuracy across all groups. If a model performs better or worse than average for a group, we consider it to be biased, showing favour for or prejudice against that group. Both favouritism and prejudice increase bias.

Group design

See Section 6.3.2.

Metadata approximation

See Section 6.3.2.

Disaggregated base metrics

Calculate performance metrics across groups.

The pattern is used as is, and includes the following sections:

1. Overview of metrics

---

1. Overview of metrics

We compared five different base metrics to evaluate model accuracy across groups: Cohen's kappa coefficient, precision, recall, weighted F1 score and the Matthews Correlation Coefficient (MCC). The trends we observed in the results analysis are consistent across metrics. Thus, we only report accuracy results for the MCC, which is a robust metric for multiclass classification [40].

Group-based Bias Measure

Compare base metrics across groups or between groups and overall performance to assess whether different groups are treated equally.

The pattern is used as is, and includes the following sections:

1. Overview of bias measure

---

1. Overview of bias measure

The group-based bias measure should penalise favouritism and prejudice equally, should be able to score models as being more or less biased, and should consider positive and negative prediction outcomes. Given these requirements, model bias with respect to a group i ($i = 1 \ldots N$) is calculated with the group-to-average log ratio:

$$bias_i = ln\left(\frac{MCC_i}{MCC_{average}}\right) \tag{6.1}$$

where MCC_i is the predictive performance of data samples belonging to the i^{th} group, and $MCC_{average}$ is the predictive performance across all samples in the evaluation set. $bias_i$ is 0 when a model is unbiased towards group i, negative when it performs worse than average and positive when it performs better than average for the group. The magnitude of the measure is equal for a performance ratio and it's inverse, as $ln(x) = -ln(\frac{1}{x})$. This has intuitive appeal that supports the interpretability of the measure: $bias_i$ is equal in magnitude but has opposing signs for groups that perform half as good and twice as good as average.

Visualising metrics and measures

Plot metrics and measures across groups to visually examine performance disparities.

The pattern is used as is in Section 6.4.1.

Meta-measure

Aggregate bias measures across groups into a single bias measure for the model to compare it against other models.

The pattern is used as is, and includes the following sections:

1. Overview of meta-measure

---

1. Overview of meta-measure

Given the group-based bias measure, we compute the sum of absolute bias score values across all groups as meta-measure, which we refer to as *reliability bias* going forward:

$$reliability\ bias = \sum_{i=1}^{N} |bias_i| \tag{6.2}$$

In this study we only consider two groups ($N = 2$), which we assume to be equally important. The *reliability bias* meta-measure is thus unweighted and does not take group size into consideration. *Reliability bias* has a lower bound of 0, and an infinite upper limit. Lower scores are preferred and signify that the performance across all groups is similar to the overall performance.

6.4. Analysing the Impact of Design Choices

Guided by the *Bias Measurement* recipe laid out in the previous section, this section presents the results of our empirical study to investigate how design choices in the on-device ML workflow propagate *reliability bias*. The section first analyses the impact of design choices during model training, namely the model architecture, sample rate and pre-processing parameters. Following this, design choices during model optimisation, in particular pruning hyperparameters, are analysed.

6.4.1. Design Choices during Model Training

To analyse the impact of the pre-processing parameters, we performed factorial ANOVA tests that allow for interactions on our balanced study design. This type of statistical test is used to determine the influence of two or more independent variables on one dependent variable [212], which makes it suitable for our study. We coded deviation (or sum) contrasts and used type 3 sums of squares. The analysis was done in python using the *scipy stastmodels* package and is available as a jupyter notebook on github[3]. Given the large number of possible interactions between the independent variables (i.e. choice variables in Table 6.1), we designed the first factorial ANOVA model (see Model 1 in Appendix A) to consider a subset of interactions that we deemed important for accuracy and bias of KWS models based on prior visual analysis. We continued to improve the factorial ANOVA models separately for the two dependent variables, MCC (accuracy) and reliability bias, by removing all non-significant interactions, and then including lower-level interactions. The final ANOVA models are included in Appendix A, with Models 2 and 3 capturing variables and interactions of model training design choices on the accuracy score and reliability bias respectively.

Tables 6.4 and 6.5 show statistically significant interaction and main effects of the final factorial ANOVA models on MCC and reliability bias. For completeness we have

[3] https://github.com/akhilmathurs/fair-ondevice-ML

Factorial ANOVA main and interaction effects	SS	df	F	p(<0.05)
model architecture	31.9714	1	6.0103E+04	0.0E+00
sample rate	3.4011	1	6.3938E+03	0.0E+00
dataset	160.1572	4	7.5270E+04	0.0E+00
mfccs	17.2272	5	6.4771E+03	0.0E+00
mel filter banks	3.1283	5	1.1762E+03	0.0E+00
frame step	0.1500	2	1.4103E+02	1.8E-61
model architecture * mel filter banks	0.0202	5	7.6075E+00	3.8E-07
dataset * sample rate * mfccs	0.1425	20	1.3391E+01	7.0E-45
dataset * model architecture * mfccs	0.1056	20	9.9288E+00	3.5E-31
Residual	9.100465	17108	-	-
Model	-	171	2.5277E+03	0.0E+00
R^2**:**	0.9619			
Adjusted R^2**:**	0.9615			

Table 6.4: Significant main and interaction effects of model training design choices on **MCC (accuracy)**. SS=sum of squares, df=degrees of freedom

Factorial ANOVA main and interaction effects	SS	df	F	p(<0.05)
model architecture	0.96734	1	469.248554	1.10E-102
sample rate	0.477009	1	231.393075	6.45E-52
dataset	62.4386	4	7.5721E+03	0.0E+00
mel filter banks	0.1225	5	1.1887E+01	1.7E-11
dataset * sample rate	0.7840	4	9.5078E+01	3.9E-80
dataset * mel filter banks	0.3758	20	9.1140E+00	5.1E-28
dataset * model architecture * mfccs	0.9662	20	2.3436E+01	1.8E-85
Residual	35.4366	17190	-	-
Model	-	92	3.6795E+02	0.0E+00
R^2**:**	0.6633			
Adjusted R^2**:**	0.6615			

Table 6.5: Significant main and interaction effects of model training design choices on **reliability bias**. SS=sum of squares, df=degrees of freedom

included main effects even if they already contribute to an interaction. The final factorial ANOVA models are significant (MCC: $F(171)=2527.2$, $p=0.0$, $R^2_{adj.}= 0.9615$; reliability bias: $F(92)=367.95$, $p=0.0$, $R^2_{adj.}= 0.6615$). For reference, the critical F statistics at p-values less than 0.01 and 0.05 are shown in Table 6.6. Based on the F statistics, we reject the null hypothesis that neither design choices made during model training, nor their interactions affect KWS model accuracy and reliability bias. The R^2 values indicate that the accuracy ANOVA model ($R^2= 0.9619$, $R^2_{adjusted}= 0.9615$) captures the effects better than the reliability bias ANOVA model ($R^2= 0.6633$, $R^2_{adjusted}= 0.6615$), in which a portion of variance in the dependent variable remains unaccounted for. Next we examine the impact of the model architecture and sample rate, and of the pre-processing parameters in more detail.

df	1	2	4	5	8	10	20	40
F_{crit} **(p<0.01)**	4052.1807	98.5025	21.1977	16.2582	11.2586	10.0443	8.0960	7.3141
F_{crit} **(p<0.05)**	161.4476	18.5128	7.7086	6.6079	5.3177	4.9646	4.3512	4.0847

Table 6.6: Critical F-values for determining significance at p<0.01 and p<0.05 for different degrees of freedom (df)

Impact of Model Architecture and Sample Rate

The results of the statistical tests in Tables 6.4 and 6.5 show that model architecture and sample rate contribute to significant interaction effects that impact accuracy and reliability bias. We now examine how the base metric and meta-measure are affected by the values of choice variables and by their interactions. Figure 6.4 shows a box-plot of accuracy and reliability bias for CNN and low latency CNN (llCNN) architectures trained on 16kHz and 8kHz audio data.

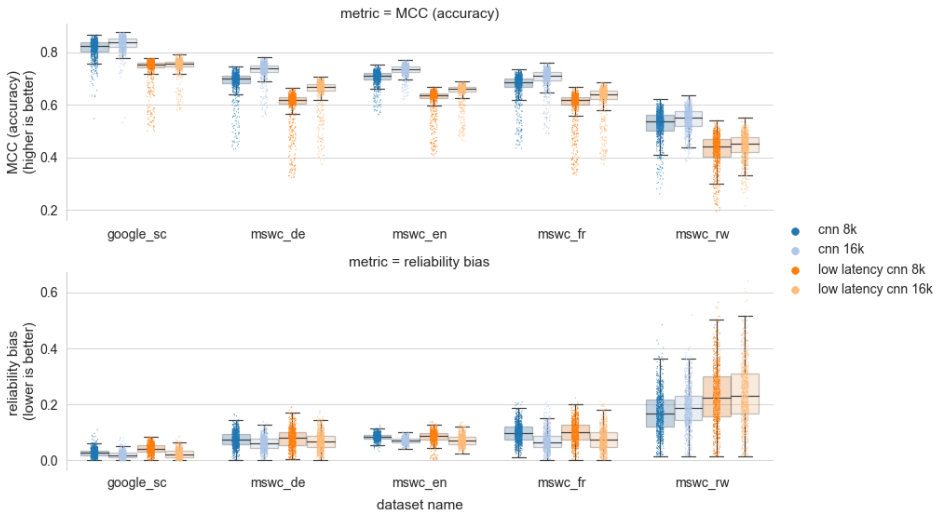


Figure 6.4: Experimental results of MCC (accuracy) and *reliability bias* for CNN and llCNN model architectures with 16kHz and 8kHz sample rates trained on 5 different datasets.

Predictive performance across architectures and sample rates

A higher MCC score implies better prediction performance. The trends in accuracy scores for models trained with different architectures and sample rates are consistent across datasets. CNN and llCNN architectures trained at 8kHz have a lower median accuracy score (i.e. they are worse) than those trained at 16kHz, and CNN architectures have higher scores than their light-weight counterparts. While models trained on the mswc_rw dataset still follow this trend, their performance, in general, is considerably worse than that of the other models. Possible reasons for this are that less training

data was available for these models, and Kinyarwanda is a different language family than the languages in the other datasets. It is out of the scope of this study to consider bias due to language and accent, which remains an important area for future work.

Reliability bias across architectures and sample rates

For reliability bias we observe that median scores are higher (i.e. worse) for models trained at 8kHz than those for models trained at 16kHz. For the google_sc, the mswc_de and the mswc_fr datasets, models trained at lower sample rates also have a higher interquartile range (IQR) in reliability bias scores. The light-weight llCNN architecture tends to have a higher median reliability bias and greater IQR than the CNN architecture, but the effect is not as pronounced as for accuracy. Models trained on the mswc_rw dataset do not follow these trends. While median reliability bias of CNN models is lower than that of llCNN models, 8kHz models are also less biased than 16kHz models. We anticipate that the deviation between trends observed for the mswc_rw models and the remaining models contributes significantly to the large effect size of the dataset variable that we observe in Tables 6.4 and 6.5.

Visualising disaggregated base metrics

Delving deeper into these findings, we analyse the relationship between male and female MCC scores across architectures and sample rates in Figure 6.5. Each data point represents the disaggregated male and female accuracy scores of a single model trained with a unique combination of pre-processing parameters. The dotted black diagonal represents equal performance for male and female speakers. Points above the diagonal perform better for females, and points below perform better for males.

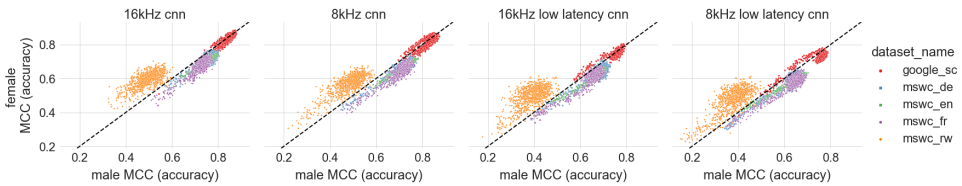


Figure 6.5: Disaggregated MCC (accuracy) scores for males (x-axis) and females (y-axis) for a single model trained with a unique combination of pre-processing parameters. On the black diagonal the performance is equal for both groups.

For the mswc_de, _en and _fr datasets it is evident that accuracy scores are biased to favour male speakers. For the mswc_rw dataset, models always favour female speakers. For the google_sc dataset the results are more nuanced. Models trained with CNN architectures tend to favour male speakers, whereas models trained with llCNN tend to favor female speakers. We also observe that for each dataset there exist models that lie on or very close to the diagonal. These models have a lower reliability bias than the remaining models. We hypothesize that pre-processing parameters contribute to reliability bias, and thus the distance of experiments from the diagonal. This leads us to

the next section, where we analyse the role of pre-processing parameters on accuracy and reliability bias.

Key insights

Model accuracy is lower at lower sample rates and for light-weight architectures. Median and IQR of reliability bias tend to be greater at lower sample rates and for light-weight architectures. The direction of bias is strongly influenced by the training dataset. Overall, male speakers are favoured by models. An exception to this are models trained on the mswc_rw dataset, which have considerably lower accuracy and favour female speakers.

Impact of Pre-processing Parameters

Having studied the effect of the architecture and sample rate, we now turn to pre-processing parameters, the next design choice listed in Table 6.1. The F statistics and p-values in Tables 6.4 and 6.5 indicate that the dimensions of log Mel spectrograms and MFCC features significantly affect accuracy and reliability bias. For accuracy there exist interaction effects between Mel filter banks and architecture, between MFCCs, dataset and sample rate, and between MFCCs, dataset and architecture. The latter interaction effect also exists for reliability bias, as well as an interaction effect between Mel filter banks and dataset.

Analysing the impact of MFCC features

Figure 6.6 visualises accuracy and reliability bias for the six MFCC and log Mel spectrogram dimensions across all datasets for the 8kHz llCNN models. Models with no MFCCs (i.e. # MFCCs = None) use only log Mel spectrograms as input features. As highlighted earlier, the lower sample rate and light-weight architecture result in models that experience greater decline in accuracy and reliability bias. We thus anticipate that the impact of pre-processing parameters is more pronounced for these models. It is clear from the figure that the accuracy of models trained with log Mel spectrograms (i.e. the blue boxes) is significantly worse than that of models trained with MFCC input features. For models trained with MFCC features, fewer dimensions (i.e. # MFCCs = 10 or 11) tend to result in a higher median accuracy than more dimensions. However, the impact of this is much smaller than that of using log Mel spectrograms. For reliability bias we observe mixed results that depend on the training dataset. Google_sc, mswc_en and mswc_rw have a lower median reliability bias when using log Mel spectrograms. On the other hand, for the mswc_de and mswc_fr datasets the median reliability bias is lower for models trained with MFCC input features.

Figure 6.7 shows comparable results for 16kHz CNN models. Here we still observe that the median accuracy is lower for log Mel spectrogram input features, except for the google_sc dataset. This dataset also has a lower median and smaller IQR of reliability bias scores when using log Mel spectrograms. Overall, the impact of the number of

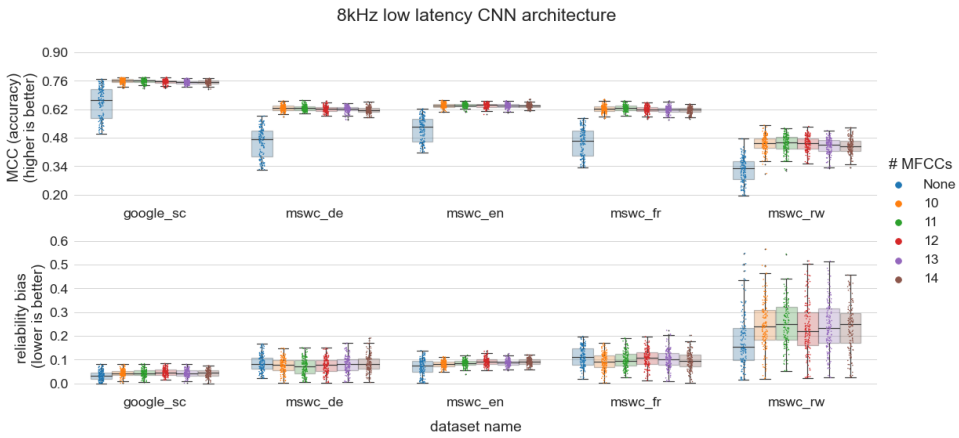


Figure 6.6: Effect of MFCC dimensions on accuracy and reliability bias for **8kHz llCNN** models. Models without MFCC features (blue), i.e. models that directly use log Mel spectrograms as input features, perform considerably worse than those that use MFCC features.

MFCC dimensions and by association the input feature type is less pronounced for CNN models trained at 16kHz.

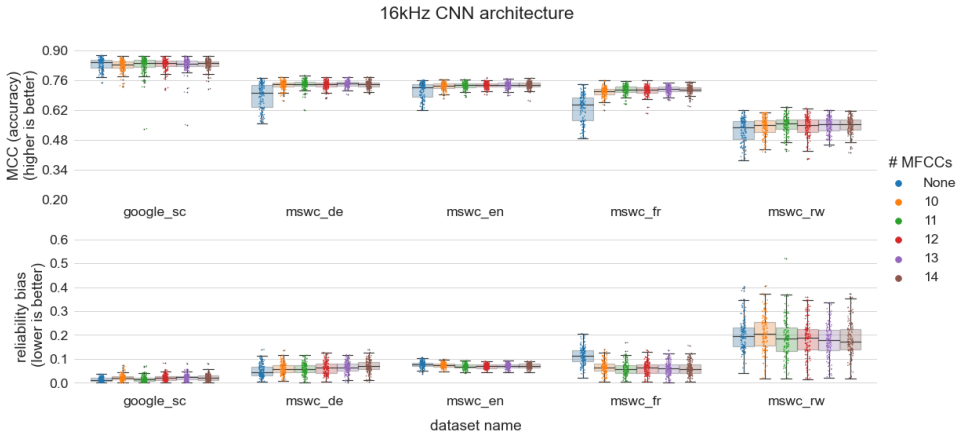


Figure 6.7: Effect of MFCC dimensions on accuracy and reliability bias for **16kHz CNN** models.

Analysing the impact of log Mel spectrogram features

Figure 6.8 visualises the impact of the number of Mel filter banks on accuracy and reliability bias for llCNN architectures. Figure 6.9 visualises results for CNN architectures, which show similar trends. It is clear that when models use log Mel spectrograms directly as input features (left hand side), the number of Mel filter bank dimensions has a critical impact on accuracy: MCC scores deteriorate rapidly as the number of Mel filter banks increases. The impact on reliability bias is more varied. For the mswc_en and

_fr datasets the Mel filter bank dimensions pose a trade-off between accuracy (models with more filter banks are less accurate) and reliability bias (models with more filter banks are less biased). Models trained with the google_sc and mswc_de datasets show no clear trend. Only models trained with the mswc_rw dataset have lower reliability bias for fewer Mel filter banks, thus allowing developers to choose Mel filter bank dimensions that increase accuracy while reducing bias.

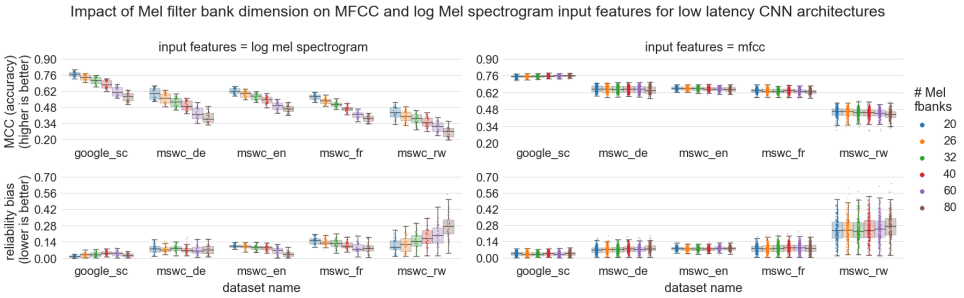


Figure 6.8: Effect of log Mel spectrogram dimensions (# Mel fbanks) on accuracy and bias, disaggregated by input feature type for **llCNN** architectures.

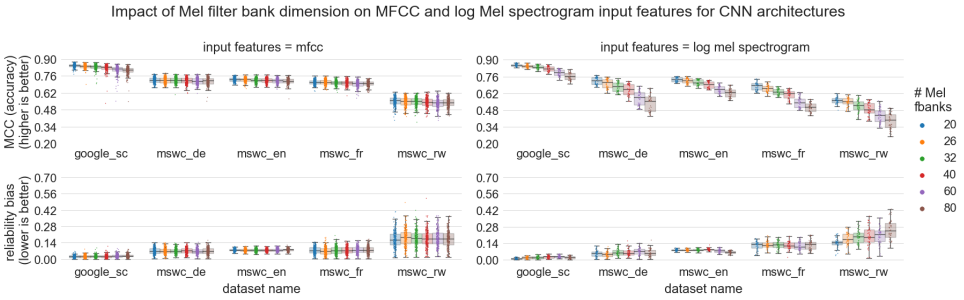


Figure 6.9: Effect of log Mel spectrogram dimensions (# Mel fbanks) on accuracy and bias, disaggregated by input feature type for **CNN** architectures.

When used with MFCCs, log Mel spectrograms serve a purpose of dimensionality reduction. In contrast to log Mel spectrogram input features, MFCC features (right hand side) are robust to the number of Mel filter banks used across all datasets. Interestingly, when comparing the results of the llCNN and CNN models trained with MFCCs in Figures 6.8 and 6.9, the distributions of accuracy scores for the google_sc and mswc_en datasets have a smaller IQR for the light-weight architecture. This suggests that the dimensionality reducing effect of the spectrograms can be particularly advantageous for smaller model architectures. As fewer input dimensions reduce the computational overhead during training and inference, our results present an opportunity for on-device ML developers: MFCCs that use log Mel spectrograms with fewer filter banks (e.g. 20) can improve computational efficiency without compromising accuracy or reliability bias.

Visualising disaggregated base metrics across feature types

To gain an appreciation of how pre-processing parameters affect the performance of KWS models for male and female subgroups, we show the impact of feature type on male and female subgroup accuracy in Figure 6.10. For mswc_de, _en and _fr accuracy is always greater for males, irrespective of the feature type. For mswc_rw the opposite holds true: accuracy is almost always greater for females, irrespective of the feature type. For the google_sc dataset log Mel spectrograms train models that have higher accuracy for females than for males, while MFCC features train models with lower accuracy for females than males. For the MSWC datasets MFCC features clearly result in more accurate models for both subgroups while for the google_sc dataset both feature types train models with similar maximum accuracy. When training on this dataset the choice of feature type can thus be a source of reliability bias.

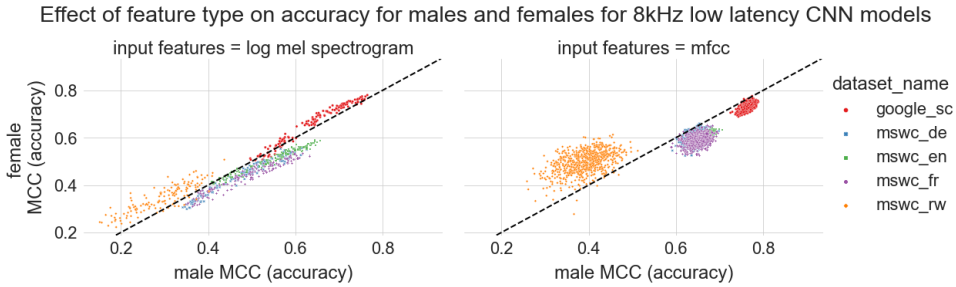


Figure 6.10: Accuracy scores for males (x-axis) and females (y-axis) for log Mel spectrogram (left) and MFCC (right) feature types for **8kHz llCNN** models.

A recent study in the speaker recognition domain highlighted that only limited feature extractors have been studied since the adoption of deep neural networks for speech processing tasks [176]. Some prior studies have noted the limits of log Mel and MFCC based features, and have proposed alternatives. Per-channel energy normalization features have been studied for robust KWS [313] and power-normalized cepstral coefficients for robust speech recognition [149]. However, while these studies consider robustness in noisy and far-field environments, they do not consider bias. Based on our findings we consider further characterisation of the effects of input features on reliability bias across a wider range of feature extractors important future work.

Key insights

Feature type and dimensions impact KWS accuracy and reliability bias. Their effect is further influenced by the training dataset. In general, MFCC type features perform better than log Mel spectrograms. However, they can also increase reliability bias, prejudicing models against females and favouring males. For MFCC features, fewer dimensions (i.e. cepstral coefficients and Mel filter banks) can reduce computational demands with a negligible impact on accuracy and reliability bias.

6.4.2. Design Choices during Model Optimisation

We focused our study of model optimisation design choices on model compression and in particular model pruning. Pruning increases model sparsity, which reduces storage, memory and bandwidth requirements when downloading models to devices. Following the experiment design and training details described in Section 6.3, we pruned a subset models with high accuracy and low reliability bias. The model selection strategy is described in detail in Section 6.5.1.

To analyse the impact of the pruning hyper-parameters, we performed factorial ANOVA tests to determine the effects of pruning hyper-parameters on change in reliability bias and change in accuracy due to pruning. The factorial ANOVA tests were designed following the same process as described for pre-processing parameters in Section 6.4.1. The first factorial ANOVA model (see Model 4 in the Appendix) considers interactions between all the independent variables, including pruning hyper-parameters, dataset, architecture, sample rate, the baseline model accuracy and baseline model reliability bias. We continued to improve the factorial ANOVA models separately for change in accuracy and change in reliability bias by removing non-significant interactions, and then including lower-level interactions. The final ANOVA models are included in the Appendix, with Models 5 and 6 capturing variables and interactions of pruning design choices on the change in MCC score and change in reliability bias respectively.

Tables 6.7 and 6.8 show statistically significant interaction and main effects of the final factorial ANOVA models. For completeness we have included main effects even if they already contribute to an interaction. The final factorial ANOVA models are significant (change in MCC: $F(274)=555.74$, $p=0.0$, $R^2_{adj.}= 0.9259$ and change in reliability bias: $F(148)=110.70$, $p=0.0$, $R^2_{adj.}= 0.5717$). We again point the reader to Table 6.6 for reference of the critical F statistics at p-values less than 0.01 and 0.05. Based on the F statistics, we reject the null hypothesis that KWS model accuracy and reliability bias are unaffected by pruning hyper-parameters and their interactions during model optimisation. As with the statistical analysis of the pre-processing parameters, we found that the R^2 values of the change in accuracy ANOVA model ($R^2= 0.9276$, $R^2_{adj.}= 0.9259$)

Factorial ANOVA main and interaction effects	SS	df	F	p(<0.05)
pruning schedule	6.2660	1	2.9467E+03	0.0E+00
bias baseline model	0.7614	1	3.5805E+02	1.1E-78
dataset	6.4134	4	7.5402E+02	0.0E+00
pruning learning rate	89.6026	2	2.1069E+04	0.0E+00
final sparsity	143.5794	5	1.3504E+04	0.0E+00
dataset * pruning schedule * final sparsity	0.2113	20	4.9694E+00	1.9E-12
sample rate * pruning learning rate * final sparsity	0.2759	10	1.2974E+01	7.2E-23
dataset * pruning learning rate * pruning schedule	0.1467	8	8.6237E+00	8.5E-12
dataset * pruning learning rate * final sparsity	2.7539	40	3.2377E+01	3.7E-232
model architecture * pruning learning rate * pruning schedule * final sparsity	0.2685	10	1.2625E+01	3.6E-22
dataset * model architecture * sample rate * final sparsity	0.1929	20	4.5357E+00	6.2E-11
Residual	25.2897	11893	-	-
Model	-	274	5.5574E+02	0.0E+00
R^2	0.9276			
Adjusted R^2	0.9259			

Table 6.7: Significant main and interaction effects of pruning hyper-parameters on **change in MCC (accuracy)**. SS=sum of squares, df=degrees of freedom

Factorial ANOVA main and interaction effects	SS	df	F	p(<0.05)
model architecture	2.0088	1	2.2874E+02	3.3E-51
pruning learning rate	15.0129	2	8.5475E+02	0.0E+00
final sparsity	23.8760	5	5.4374E+02	0.0E+00
bias baseline model	8.5336	1	9.7171E+02	3.3E-205
dataset	22.9815	4	6.5421E+02	0.0E+00
pruning schedule * final sparsity	1.0450	5	2.3798E+01	6.8E-24
model architecture * final sparsity	2.8381	5	6.4633E+01	8.5E-67
dataset * pruning learning rate * final sparsity	10.8173	40	3.0794E+01	3.2E-220
dataset * model architecture * sample rate * pruning learning rate	1.1197	8	15.937276	1.3E-23
Residual	105.508229	12014	-	-
Model	-	148	1.1070E+02	0.0E+00
R^2	0.5769			
Adjusted R^2	0.5717			

Table 6.8: Significant main and interaction effects of pruning hyper-parameters on **change in reliability bias**. SS=sum of squares, df=degrees of freedom

indicate that this model captures the effects better than the change in reliability bias ANOVA model ($R^2 = 0.5769, R^2_{adj.} = 0.5717$). In the latter model a portion of the variance in the dependent variable remains unaccounted for.

Impact of Pruning Hyper-Parameters

Next, we examine the impact of pruning design choices in greater detail. We first analyse the impact of final sparsity and pruning schedule, then the impact of final sparsity and pruning learning rate, and finally interaction effects with pruning learning rate. Throughout the analysis we use the terms *change in* and *delta* interchangeably.

Analysing the impact of final sparsity and pruning schedule

The interaction effect between final sparsity and pruning schedule is visualised in Figure 6.11. This interaction significantly affects change in bias due to pruning (as per

Table 6.8). The interaction between final sparsity, pruning schedule and dataset also has a significant effect on change in accuracy (as per Table 6.7).

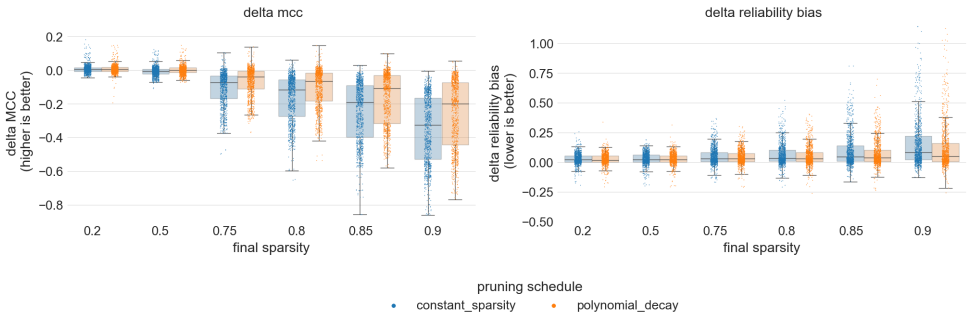


Figure 6.11: Interaction effect of final sparsity and pruning schedule on **change in MCC (accuracy)** (left) and **change in reliability bias** (right). The *polynomial decay* (orange) pruning schedule results in higher median change in MCC scores and lower median change in reliability bias. It also results in smaller reliablity bias IQR. These effects become more significant as final sparsity increases.

The figure highlights several interesting observations. When final sparsities are low (i.e. 20% and 50%), the median delta MCC and delta reliability bias are close to zero. Furthermore, the delta MCC and delta reliability bias IQR of models pruned to these sparsities are small. This indicates that these pruned models have low variability in accuracy and reliability bias and that scores lie close to those of the baseline models. However, as the final sparsity increases, the median delta MCC becomes more negative (implying lower accuracy due to pruning) and the IQR increases (indicating greater variability in accuracy due to pruning). Likewise, the median delta reliability bias and the IQR increase, indicating that models become more biased and that reliability bias scores become more variable. For all sparsities there are some models that have a positive change in MCC, thus becoming more accurate, and a negative change in reliability bias, thus becoming less biased, due to pruning. The polynomial decay pruning schedule results in higher median delta MCC scores and lower median delta reliability bias. Polynomial decay also results in smaller IQR of delta reliability bias. These effects become more apparent as final sparsity increases. For developers polynomial decay is thus a more robust pruning schedule to choose.

Analysing the impact of final sparsity and pruning learning rate

Figure 6.12 visualises the interaction effect of final sparsity and the pruning learning rate. As shown in Table 6.8 this interaction significantly affects the change in reliability bias due to pruning. Final sparsity and pruning learning rate also have a significant effect on change in accuracy through interactions with sample rate and with dataset (see Table 6.7). At a low final sparsity of 20% the learning rate has no impact on the accuracy and bias of pruned models. As the sparsity increases, this changes dramatically. The smaller the learning rate, the lower the median delta MCC and the larger its

IQR. A lower delta MCC results in a greater accuracy drop due to pruning. Similarly, the smaller the learning rate, the higher the median delta reliability bias and the larger its IQR. A higher delta reliability bias increases the reliability bias due to pruning. At 90% sparsity the median MCC score of models pruned with a learning rate of 0.00001 reduces by more than 0.5 (maximum value of the MCC metric is 1). This means that the accuracy of pruned models with 90% sparsity is less than half that of baseline models. At the same final sparsity and learning rate the median delta reliability bias increases by 0.18, indicating that substantial performance discrepancies exist between male and female groups.

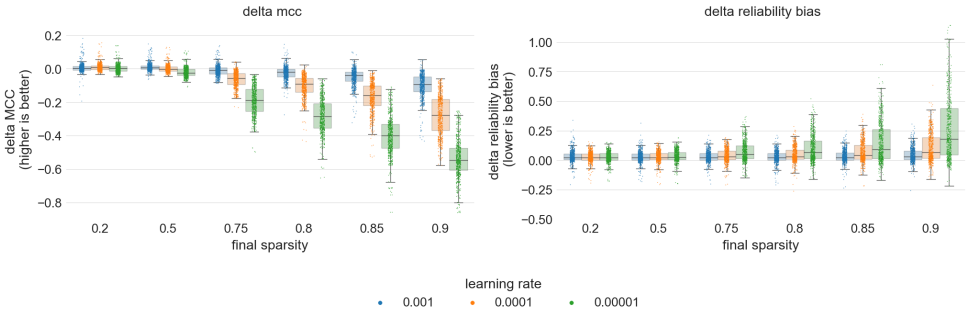


Figure 6.12: Interaction effect of final sparsity and pruning learning rate on **change in MCC** (left) and **change in reliability bias** (right). At final sparsities above 50%, smaller learning rates significantly reduce MCC scores and increase reliability bias.

A possible explanation for our results is that the learning rate optimises the discovery of structure in the training data to favour one group over the other. This intuition aligns with recent empirical work that shows that top performing deep neural networks can have very similar accuracy, but large variance in other performance aspects such as inference latency due to hyper-parameter tuning [171]. Similarly, a recent study on fixed-seed training of deep learning systems shows high variance in fairness measures if experiments consist of a single run with a fixed seed [235]. Based on our results, developers can choose a larger pruning learning rate, like 0.001, during model optimisation to reduce the likelihood of unintended bias and unexpected accuracy degradation, especially when pruning models to high sparsities. While this rule-of-thumb is useful given our current knowledge, further research is needed to fully characterise the impact of the pruning learning rate on model performance and bias. We thus suggest that developers empirically validate and optimise the learning rate during pruning.

Analysing interaction effects with pruning learning rate

To conclude our detailed analysis of interaction effects arising during model pruning, we examine how the interactions between dataset, architecture, sample rate and pruning learning rate affect change in reliability bias in Figure 6.13. Across datasets, archi-

tectures and sample rates we observe the general trend which we have already identified in the previous figure: the smaller the learning rate, the more biased models become. Careful examination of the results across architectures and sample rates also reveals trends similar to those we observed with pre-processing parameters: the increase in reliability bias due to pruning is greater for the light-weight low latency CNN architecture and the lower sample rate of 8kHz, as indicated by higher medians and larger IQRs. This trend is stronger at smaller learning rates. As with the pre-processing parameters, the mswc_rw dataset presents an exception to this observation. For this dataset median delta reliability bias across learning rates shows no clear trend, while the IQRs are always large when compared to the IQRs of the other datasets.

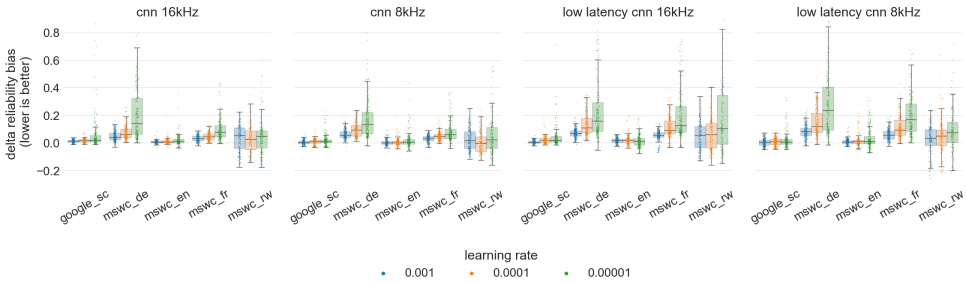


Figure 6.13: Interaction effect between dataset, architecture, sample rate and pruning learning rate on **change in reliability bias**. Across datasets the general trend indicates that the smaller the learning rate, the more biased models become. The median change in reliability bias of the google_sc and mswc_en datasets is less affected by the pruning learning rate than that of the remaining datasets

Figure 6.13 reveals a further insight when comparing results across datasets. The reliability bias of models trained on the google_sc and mswc_en datasets is less affected by the pruning learning rate than what is the case for models trained on the remaining datasets. These two English language datasets have low median delta reliability bias values and a small IQR. A likely contributor to these results is that English is the best resourced language, with the largest available quantity of data. The English datasets in our study thus include more utterances per keyword, more unique speakers per keyword and better representation of speakers and utterances across keywords in the validation and test sets.

Data quantity and representativeness, however, may not explain the entire effect. The mswc_de and mswc_fr datasets have very similar statistics across the keywords, genders and dataset splits, with the mswc_de dataset being 13% larger than the mswc_fr dataset. Yet, the change in reliability bias for mswc_de models is larger and more variable than that of mswc_fr models. Further research is needed to understand the source of this variability. For developers, our results highlight that training, validation and test datasets need to be large enough and representative across groups of users to ensure robust results and avoid bias. Considering that German and French are, after English,

two of the best resourced languages in the Mozilla Common Voice corpus, this highlights the need to collect context and application specific datasets to train models and evaluate bias.

Key insights

Polynomial decay is a more robust pruning schedule than constant sparsity, and a larger pruning learning rate, like 0.001, reduces the likelihood of unintended bias and unexpected accuracy degradation. These design choices are particularly important when pruning models to sparsities greater than 50%, beyond which accuracy and reliability bias can deteriorate dramatically. The increase in reliability bias due to pruning is greater for smaller architectures and at lower sample rates. This trend is stronger at smaller learning rates. Training, validation and test datasets need to be large enough and representative across groups of users to ensure robust results and avoid bias.

6.5. Mitigating Bias in On-device KWS

The previous section reported empirical results and an analysis of the impact of design choices on accuracy and reliability bias for an audio KWS task. Taking the insights gained through the study into consideration, this section introduces two patterns that present low effort strategies for mitigating bias in on-device ML. Next, the section introduces a pattern for model selection and then one for data-driven design decisions to inform model development decisions with targeted experimentation. The section concludes with a validation of the new patterns against the requirements of Chapter 4.

6.5.1. Model Selection

Rather than considering reliability bias and accuracy as a trade-off, engineers should seek approaches to navigate multi-objective search scenarios where high accuracy and low reliability bias are desired properties of a model. Bias-aware model selection can be formulated as a new pattern.

New pattern: Bias-aware Model Selection

Use a multi-objective criterion that considers accuracy and bias to select models that have high accuracy and low bias after training or optimisation.

In the decision map presented in Figure 6.3 model selection can occur after training a new model, downloading pre-trained models or optimising a model. In contrast to the *In-processing model selection* pattern that imposes multi-objective criteria during model training [175, 220], the *Bias-aware model selection* pattern focuses on

multi-objective model selection as a post-processing intervention. It is thus a post-processing pattern used for the purpose of design. Next, the pattern is instantiated after model training and then after model optimisation to demonstrate how model selection strategies that account for accuracy and reliability bias have been instantiated in our empirical study.

Model Selection After Training

Section 6.4.1 explored how model training design choices impact reliability bias and accuracy. While the analysis highlighted trends across datasets and architectures, we also found that models exist that are accurate and that perform equally well for male and female groups. A visual appreciation for this can be gained from Figure 6.5, where models that lie on or close to the diagonal have equal accuracy for males and females. Such models occur across datasets, architectures and sample rates, suggesting that pre-processing parameters exist that produce models with high accuracy and low bias. However, these models do not necessarily have the highest accuracy score.

Model selection criteria

An alternative to selecting the model with the best accuracy is to consider search criteria for selecting models based on accuracy and reliability bias. Listed below are three criteria we used to select model d from n trained models D by optimising for:

1. **high accuracy**:
   select d if $MCC_d = max(MCC_1, ..., MCC_n)$ for n in D
2. **low bias**:
   select d if $reliability\ bias_d = min(reliability\ bias_1, ..., reliability\ bias_n)$ for n in D
3. **low bias + high accuracy**:
   select d if $MCC_d >= 0.985 * max(MCC_1, ..., MCC_n)$ for n in D **and** $reliability\ bias_d = min(reliability\ bias_1, ..., reliability\ bias_k)$ for k where $MCC_k >= 0.985 * max(MCC_1, ..., MCC_n)$ for n in D

We considered the high accuracy criteria as a baseline, as this is the typical strategy followed by engineers that do not consider bias. The low bias criteria presents the opposite scenario, where only reliability bias informs model selection. Finally, the low bias + high accuracy criteria considers accuracy as a satisficing metric while minimising reliability bias. This criteria selects the model with the lowest reliability bias, provided that it has an accuracy score within a 1.5% threshold of the maximum accuracy. A reasonable threshold value should be selected in accordance with the application requirements.

Results analysis across selection criteria after training

This multi-objective approach allowed us to explore alternative models for deployment. Table 6.9 shows the MCC score and reliability bias for the best models trained on the google_sc dataset, selected according to the three criteria. The low bias + high accuracy criteria selects models with a low reliability bias across architectures, while

retaining an MCC score close to the high accuracy criteria. For the CNN architectures, this criteria reduces reliability bias by 15.7 and 1.7 fold for models trained with 16kHz and 8kHz sample rates respectively. For the 8kHz llCNN model, reliability bias is reduced 22.3 fold. For the 16kHz llCNN architecture the model with the highest accuracy also has the lowest reliability bias and the measure thus remains unchanged. By comparison, models selected using only low bias as selection criteria have a very low reliability bias. However, this comes at the cost of an accuracy drop between 3.2% and 6.1%, which is considerably greater than the desired 1.5% threshold. This selection criteria will adversely affect both groups and degrade their performance.

model selection criteria	metric	16kHz CNN	8kHz CNN	16kHz low latency CNN	8kHz low latency CNN
high accuracy	MCC score	0.877	0.868	0.804	0.778
	reliability bias	1.2e-2	9.8e-3	6.6e-4	4.1e-2
low bias	MCC score	0.849	0.815	0.762	0.740
	reliability bias	1.8e-4	1.9e-4	1.2e-4	1.6e-4
low bias + high accuracy	MCC score	0.872	0.861	0.804	0.775
	reliability bias	7.7e-4	5.9e-3	6.6e-4	1.8e-3

Table 6.9: Table of MCC scores and reliability bias for models trained on the google_sc dataset and selected for 1) high accuracy 2) low bias 3) low bias + high accuracy. The low bias + high accuracy criteria selects models with considerably lower bias than the high accuracy strategy.

Model selection for pruning experiments

Instead of selecting maximum or minimum values, the selection criteria can be modified to select the m best models under that criteria. We followed this approach choosing $m = 3$ best models for a dataset, model architecture and sample rate triplet to select baseline models for the pruning experiments. The low bias + high accuracy criteria did not return valid models for all triplets, which is the reason for the unequal number of baseline models in Table 6.3. Next we consider how these selection criteria hold up after pruning.

Model Selection After Pruning

Change in accuracy and bias across selection criteria

Figure 6.14 shows density distributions of delta (i.e. change in) MCC score (top) and delta reliability bias (bottom) after pruning, for models selected under the three criteria after training. For the density distributions, accuracy increases in the direction of positive change, meaning that delta MCC distributions that peak to the right of zero are desirable. Conversely, reliability bias decreases in the direction of negative change, meaning that delta reliability bias distributions with peaks to the left of zero are desirable. In the figure, the delta MCC distributions peak just left of zero (CNN) or on zero (llCNN), indicating that the majority of models with these architectures experience a

decline in accuracy. The shape of the distributions is similar for different selection criteria under the same architecture and sample rate, with the accuracy of low bias models (which have lower baseline accuracy) increasing slightly more after pruning.

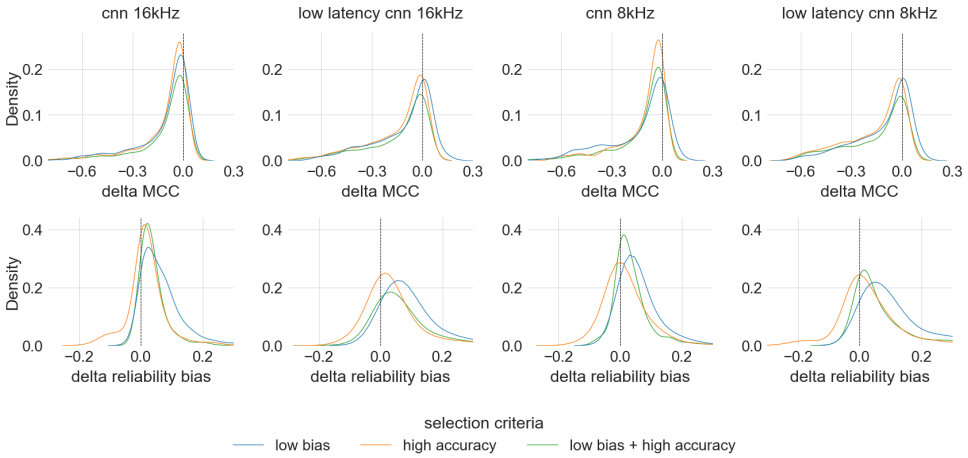


Figure 6.14: **Change in MCC** score and **change in reliability bias** after pruning. Models were selected for 1) high accuracy (orange), 2) low bias (blue), and 3) low bias + high accuracy (green). Delta MCC is better when greater, delta reliability bias is better when smaller. The selection criteria affects delta reliability bias but not delta MCC.

The shapes and peaks of the delta reliability bias distributions vary across model selection criteria. This indicates that the model selection criteria impact reliability bias after pruning. A further confirmation of this is presented in the statistical analysis in Table 6.8, where the reliability bias of the baseline model has a statistically significant effect on delta reliability bias due to pruning. Analysing the distributions, we can see that the distributions of models selected for low bias (blue) lie furthest to the right. This means that the reliability bias of these models increases the most after pruning. This is not surprising, as the lower bound of the reliability bias measure is zero and models with low bias are very sensitive to small changes in reliability bias. However, this can also indicate that models selected for low bias may loose some of their good bias properties during pruning. The distributions of models selected for high accuracy (orange) lie furthest to the left. These models typically started out with higher reliability bias after training, which makes them less sensitive to changes in reliability bias and thus better able to retain their reliability bias scores.

Model properties across selection criteria

Figure 6.15 visualises the distribution of MCC scores and reliability bias for the selection criteria. Right of the peak (i.e. in the higher accuracy range), the MCC score distributions for the high accuracy criteria and the low bias + high accuracy criteria lie very close to each other. After pruning, models selected for high accuracy and for low

bias + high accuracy thus have similar MCC scores. For reliability bias the distribution of the low bias + high accuracy criteria lies between the low bias and high accuracy distributions. The low bias + high accuracy criteria thus results in models with lower bias after pruning than the high accuracy criteria. Overall, this makes the low bias + high accuracy criteria a good choice to select a range of models for pruning.

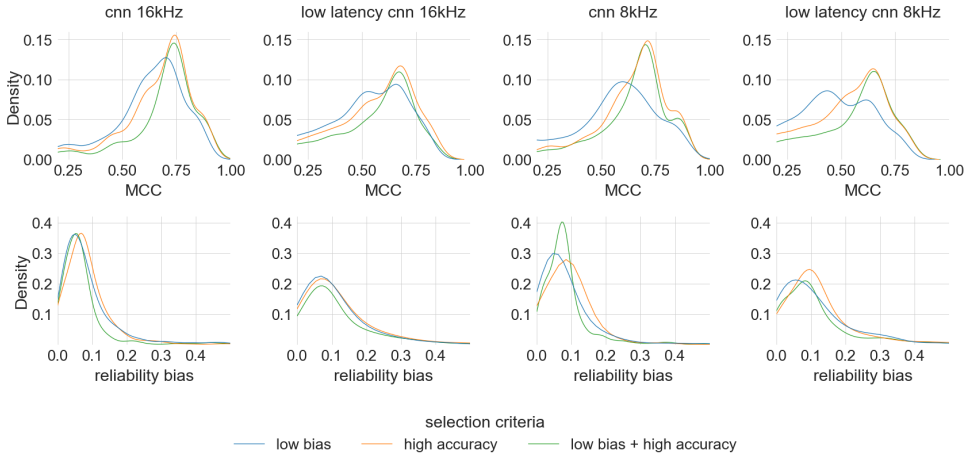


Figure 6.15: **MCC** score and **reliability bias** of models after pruning. Models were selected for 1) high accuracy (orange), 2) low bias (blue), and 3) low bias + high accuracy (green). After pruning MCC is greatest for models selected with a criteria that considers high accuracy. Similarly, reliability bias is lower for criteria that consider low bias.

Results analysis across selection criteria after pruning

Finally we reapplied the same model selection criteria that we previously applied after training, *after pruning*. Table 6.10 shows the mean and standard deviation of accuracy and reliability bias across sparsities for the three selection criteria for *pruned* models trained on google_sc. Mean reliability bias can be improved by an order of magnitude by choosing the low bias + high accuracy criteria rather than the high accuracy criteria. While the low bias criteria results in lower reliability bias than the low bias + high accuracy criteria, the latter already has a low mean and variance in reliability bias, making additional reductions less impactful. Models selected with the low bias criteria, on the other hand, suffer a large drop in accuracy. For all models the variance of metrics across sparsities is relatively low, which supports our earlier observation that models trained on the google_sc dataset are less affected by pruning hyper-parameters than models trained on other datasets (see Figure 6.13).

Across all datasets the low bias + high accuracy criteria selects models with similar accuracy and lower reliability bias than the high accuracy criteria. This outcome is not surprising, as the purpose of a multi-objective criterion is precisely to satisfy multiple objectives. The value of this analysis lies in empirically validating the obvious rather

criteria	high accuracy				low bias + high accuracy				low bias			
metric	MCC score		*reliability bias*		MCC score		*reliability bias*		MCC score		*reliability bias*	
	mean	var	mean	var	mean	var	mean	var	mean	var	mean	var
16kHz CNN	0.885	1.2e-02	1.4e-02	9.6e-03	0.879	1.1e-02	4.0e-03	5.5e-03	0.823	5.3e-02	6.5e-04	4.0e-04
8kHz CNN	0.876	9.2e-03	6.5e-03	1.8e-03	0.870	8.9e-03	1.1e-03	6.7e-04	0.851	1.9e-02	2.9e-04	2.6e-04
16kHz llCNN	0.808	1.8e-02	8.9e-03	7.0e-03	0.804	1.7e-02	1.4e-03	1.1e-03	0.772	2.3e-02	4.9e-04	4.1e-04
8kHz llCNN	0.785	2.5e-02	1.0e-02	7.6e-03	0.781	2.4e-02	1.8e-03	2.1e-03	0.761	3.5e-02	4.9e-04	4.9e-04

Table 6.10: Mean and variance of MCC scores and reliability bias across all pruning sparsities for the three model selection criteria. Models have been trained on the google_sc dataset.

than in finding surprise: engineers can reduce bias in audio KWS with little effort by using the *Bias-aware model selection* pattern and choosing models that satisfy an accuracy condition while minimising bias.

Summary of instantiated *Bias-aware model selection* pattern

Engineers should use a multi-objective criterion that considers accuracy and bias to select models that have high accuracy and low bias after training or after pruning. We propose that engineers set a tolerance that controls the drop in accuracy from the maximum value, thus using accuracy as a satisficing metric while minimising bias. The tolerance value should be determined from application requirements. If model training is followed by pruning, a small number of top models should be selected for pruning using high accuracy and low bias + high accuracy strategies.

6.5.2. Targeted Experimentation to Inform Design Decisions

Section 6.3 presented a map of design choices arising in the on-device ML workflow. We then showed empirically that these design choices can lead to disparate performance of audio KWS models for males and females. The analysis in Section 6.4 demonstrated that even when engineers make reasonable decisions about training and optimisation parameters (see Table 6.1) their choices can lead to models with widely different accuracy and bias properties. Especially when training light-weight architectures or processing data at low sample rates, systematic experimentation is thus a necessary strategy to support design decisions and mitigate bias. However, experimentation comes at a cost: each iteration requires computational resources, takes time and consumes energy. Engineers should take the resource footprint and cost of model training into account, and target their experiments to iterate over values that are likely to yield high accuracy, low bias models. These observations have informed the formulation of a new pattern.

New pattern: Data-driven Design Decisions

Support design decisions with targeted and systematic experimentation while evaluating bias and accuracy metrics.

The *Data-driven design decisions* pattern calls for targeted experimentation to enable

data-driven decision making throughout the ML development workflow. It is thus an interaction pattern with the purpose of design.

Revised design choice variables for audio KWS

We have demonstrated that iterating over pre-processing parameters during training, and pruning hyper-parameters during model compression can help engineers train models with high accuracy and low bias. However, the empirical experiments were resource intensive and a costly undertaking. Where a single audio KWS model takes only a couple of minutes to train, we trained 17280 models, pruned 12168 models and ran experiments for several days. Rather than replicating our empirical approach for all design choice variables, experiments should be targeted.

Design action	Design choice	Choice variable (unit)	Variable values
Train new model	input features | sample rate	sample rate (kHz)	*determined by application*
Train new model	input features | pre-processing	feature type	MFCC
Train new model	input features | pre-processing	# Mel filter banks	20, 32
Train new model	input features | pre-processing	# MFCCs	10, 11
Train new model	input features | pre-processing	frame length (ms)	20, 25, 30, 40
Train new model	input features | pre-processing	frame step (% frame length)	40, 50, 60
Train new model	input features | pre-processing	window function	Hamming
Optimise model	light-weight architecture	model architecture	*determined by application*
Optimise model	model compression | pruning	final sparsity (%)	*determined by application*
Optimise model	model compression | pruning	pruning frequency	10, 100
Optimise model	model compression | pruning	pruning schedule	polynomial decay
Optimise model	model compression | pruning	pruning learning rate	1e-4, 1e-5 for sparsities < 50%; 1e-3 for sparsities > 50%
Model selection	selection strategy	criteria	high accuracy, low bias + high accuracy
Model selection	selection strategy	# best models	3

Table 6.11: Recommended design choice variables and values for audio KWS to mitigate bias while reducing resource consumption during experimentation

To this end Table 6.11 revises design choice variable values based on the insights we gained through the experiments. Given these reduced options, engineers only need to train 48 models per sample rate and architecture (instead of 864), and run at most 24 pruning experiments for low sparsities (12 experiments for higher sparsities of more than 50%). This targeted approach reduces resource consumption by 95%, which has a significant impact on future training time and energy requirements. Such a targeted and systematic approach to experimentation is necessary for time sensitive and resource constrained projects, and makes it more feasible for engineers to use data-driven decision making to mitigate bias in on-device ML workflows.

6.5.3. Validating New Patterns

The patterns are formulated as *name/key idea* pairs, are unique, and formulated as simply as possible without loosing contextual nuance. They are meaningful on their own

and consistent with other patterns. Currently they are not included in any recipes, and make independent contributions to bias mitigation. The *Bias-aware model selection* pattern is a post-processing pattern, and the *Data-driven design decisions* is an interaction pattern. Both patterns are used for design purposed and have been instantiated in this section to investigate their utility for mitigating bias in a practical context. The new patterns thus satisfy the requirements set out in Chapter 4, though requirements 5, 9 and 11 need to be validated in future work.

6.6. Conclusion

The patterns identified and validated in Chapters 4 and 5 underwent their final develop-demonstrate-validate-design iteration in this chapter. They were extended and adapted to an Edge AI use case, where they were used to study the propagation of bias due to design choices during model training and optimisation of on-device keyword spotting (KWS) systems. The chapter thus successfully extended the patterns and processes captured by the *Benchmark Dataset* and *Bias Measurement* recipes to on-device ML, while also demonstrating their utility for conducting ongoing bias evaluations during ML development. This shows that the patterns are reusable and applicable to different application contexts, thus satisfying requirements 9 and 11. The research that informed this chapter presented the first evidence of bias in on-device ML.

The chapter introduced design choices during model development in Section 6.2, and framed them as potential sources of bias in the inherently constrained on-device ML setting. The design choices were then investigated empirically in a bias evaluation in the remainder of the chapter. Section 6.3 provided details on the study set up and the bias evaluation. The *Benchmark Dataset* recipe guided the design of KWS datasets for training and evaluation, and the *Bias Measurement* recipe guided the bias evaluation. Both recipes were used in proactive mode and remained unaltered in this design iteration.

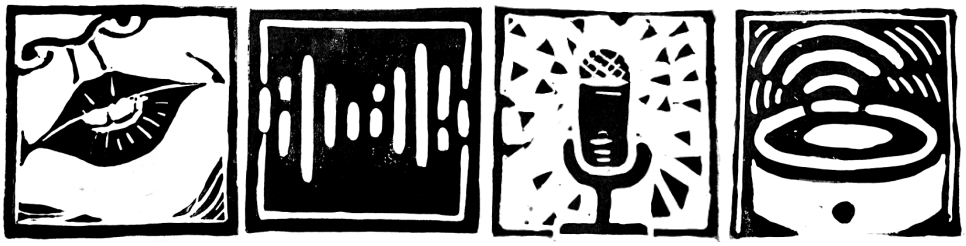
Section 6.4 showed how design choices impact accuracy and reliability bias during model training and optimisation in the audio KWS use case. During model training, design choices that relate to the data sample rate, pre-processing parameters and model architecture were investigated. During model optimisation, the investigated design choices relate to pruning for model compression. The study results showed that these design choices, made by engineers at different stages of the on-device ML workflow, can lead to disparate predictive performance across groups of users based on their gender. While some design choices may lie beyond the immediate control of engineers, they have full control over others. Section 6.5 proposed two new patterns with a design purpose that give engineers an active role in detecting and mitigating bias in on-device ML. The *Bias aware model selection* pattern is a post-processing pattern that engineers should use to ensure that models selected after training and pruning have high accuracy and low bias. The *Data-driven design decisions* pattern is an interaction

pattern that engineers should use to support their design decisions during on-device ML development with targeted and systematic experimentation.

This chapter has demonstrated through rigorous, empirical research that the patterns developed in this thesis are reusable and extensible. The pattern catalogue can be adapted to include new design knowledge specific to the Edge AI setting and the patterns in the recipes extend to this application area. This chapter thus successfully addresses Research Question 3 and accomplishes Goal 3, marking the end of the design science research cycle laid out in Figure 1.2. The next chapter discusses the key findings and limitations of the research conducted in this thesis.

7

Discussion



7.1. Overview

From smart phones to speakers and watches, Edge AI is deployed on billions of devices to process large volumes of personal data efficiently, privately and in real-time. While Edge AI applications are promising, many incidents of bias in AI systems caution that bias should also be detected and mitigated in Edge AI. However, research on bias detection and mitigation in Edge AI is extremely limited. Prior work on bias and fairness in machine learning (ML) assumes AI systems to be trained and deployed in cloud-based settings, where compute and energy resources are unconstrained. By contrast, Edge AI deploys ML models to localised edge servers and devices. This improves the privacy of data processing and reduces bandwidth constraints, data communication overheads, response latency and data transfer costs. But edge devices frequently have limited energy supply and constrained computing resources, which can adversely impact bias. Thus, to support the demand for trustworthy AI, and especially the requirement for diversity, non-discrimination and fairness, this thesis studied generalisable approaches to bias detection and mitigation during Edge AI development.

Despite the unique constraints of Edge AI, bias detection and mitigation could be supported by making established knowledge from ML fairness accessible for reuse. This experience could then be transferred to researchers and practitioners in Edge AI that use ML methods, but who have limited prior experience in bias detection and mitigation. In the literature, no previous work has attempted to capture such general knowledge for bias detection and mitigation from ML fairness. However, in software engineering, design patterns have been found a helpful tool to solve a similar problem. Design patterns capture ideas that have been useful in one context and that will probably be useful in others [184]. Their success at capturing repeating solutions to recurring problems in software engineering positions them as a promising approach for capturing and transferring knowledge on bias detection and mitigation in ML for reuse in Edge AI. **This thesis thus aimed to develop design patterns for detecting and mitigating bias in the development of Edge AI systems.**

This thesis used the Design Science Research Method of Peffers et al. [227] to develop the patterns. The iterative development of design patterns was guided by three goals and corresponding research questions. The goals and questions were aimed at gathering and extending knowledge from ML fairness by studying artifacts, through empirical investigation, and through analysis of use cases of voice-activated systems. The first question established foundational knowledge for developing initial design patterns based on best-practices in ML fairness. The second question validated the patterns for a general ML use case, and the third question extended the patterns to an Edge AI use case. The goals and questions are summarised below.

1. **Goal 1 and RQ1** sought to **identify** and capture established practices and knowledge on bias detection and mitigation in ML fairness as design patterns. The research results have been presented and discussed in detail in Chapter 4.

2. **Goal 2 and RQ2** sought to **validate** the utility of the proposed design patterns in a ML use case where bias has not been studied previously. The utility of the patterns for detecting and mitigating bias has been demonstrated and evaluated in a speaker verification use case in Chapter 5.
3. **Goal 3 and RQ3** sought to **extend** the design patterns to Edge AI applications, and to identify new knowledge regarding bias detection or mitigation in Edge AI. This was done in an on-device keyword spotting use case, which is presented in Chapter 6.

This chapter examines the extent to which the design patterns and this thesis have succeeded in capturing knowledge and practices from ML fairness as design patterns to support bias detection and mitigation in Edge AI. Next, Section 7.2 highlights the scientific and societal contributions of the thesis. The findings of the research questions and insights gained while accomplishing the accompanying goals are summarised in Section 7.3. Section 7.4 considers limitations of this work. Finally, Section 7.5 offers recommendations and suggestions for future research.

7.2. Research Contributions

This section summarises the key scientific contributions of this work and highlights its relevance to society and industry. It then presents a discussion of the contributions.

7.2.1. Scientific Contributions

1. This thesis is the first work to propose design patterns as an explicit solution for transferring established practices and knowledge on bias detection and mitigation from ML fairness to new domains.
2. Furthermore, this thesis creates patterns and recipes as design artifacts to detect and mitigate bias in Edge AI. The artifacts are created following a rigorous design science research approach and three design iterations to develop, demonstrate and validate them.
3. The design science research approach and the design artifacts that this thesis creates make a methodological contribution to research in ML fairness, demonstrating the value that established design practices hold for the development of diverse, fair and non-discriminatory AI.
4. Through validating the design patterns in the first use case, this thesis presents the first empirical and analytical analysis of bias in the end-to-end development workflow of speaker verification systems.
5. By extending the design patterns to the second use case, this work detects previously unknown sources of bias in the development of on-device keyword spot-

ting systems. To my knowledge, this study also presents the first investigation of bias in the development of on-device ML systems.

7.2.2. Relevance to Society and Industry

This thesis responds to the the societal call for non-discriminatory, fair and inclusive AI by developing practical and implementable approaches to support bias detection and mitigation in the development of Edge AI. The design patterns and recipes proposed in this thesis provide practitioners with tools to contemplate the moral impact of their work, and take proactive actions to detect and mitigate bias in the development of Edge AI systems. By applying the artifacts developed in this research in the development workflow of Edge AI products and services, corporations can avoid harmful and embarrassing product failures, and make progress towards compliance with new legislation focused on non-discrimination and fairness of AI systems[1]. By formalising bias detection and mitigation in Edge AI with design patterns and by capturing processes with pattern recipes, choices and decisions can be documented and explicitly considered in the design process. This helps to make bias evaluations coherent, reproducible and consistent, and facilitates the trustworthy AI requirement of transparency. Finally, this thesis brings to attention the urgent need to address bias in emerging and increasingly pervasive technologies that process personal data collected by the Internet of Things.

7.2.3. Discussion of Contributions

The highlighted contributions of this thesis can be considered through different lenses. Through the lens of Edge AI, the thesis adapts and extends proven techniques from research in ML fairness to the emerging challenge of bias detection and mitigation in the Edge AI domain. Specifically, the thesis developed and validated artifacts, namely individual design patterns, a pattern catalogue and two pattern recipes, to assist researchers and practitioners in Edge AI to detect and mitigate bias.

Considering that the purpose of design patterns is to capture established practices and knowledge for reuse, I consider them as nascent design theory that provides prescriptive knowledge on how to build Edge AI systems that are inclusive, non-discriminatory and fair. By specifying bias evaluations and interventions with design patterns and recipes, researchers and practitioners can make their assumptions explicit and clarify the intent behind their design choices. The artifacts are useful in that they are easy to understand, and make established practices and knowledge on bias detection and mitigation accessible. Furthermore they provide a language for clarifying common concepts, which can facilitate communication and collaboration between technical and non-technical stakeholders. These contributions advance the state of the art of trustworthy Edge AI.

[1] such as the EU AI Liability Directive and the US AI Bill of Rights

Viewed from the perspective of ML fairness, the thesis makes a contribution to the practical challenge of knowledge capture and transfer in the domain. The thesis contributes the concept of design patterns to capture key ideas, and pattern recipes to capture processes, as solutions to this challenge. The viability of the solution is demonstrated by designing a concrete instantiation of design patterns for detecting and mitigating bias in Edge AI, and rigorously validating them against requirements in two use cases. The patterns and recipes capture best practices in ML fairness in a generalisable manner, and enable people new to the domain to learn from the community's collective experience. By making proven knowledge accessible, patterns and recipes also offer a solution for repeatable and coherent bias evaluations. Through their development, this thesis demonstrates the methodological value of design approaches, in particular design science research, for addressing important problems in AI. Together, these contributions advance the state of knowledge of diverse, fair, and non-discriminatory ML, and by extension trustworthy AI.

Finally, the develop-demonstrate-validate design iterations of the speaker verification and on-device KWS use cases contribute descriptive knowledge specific to these application domains. This thesis has presented the first comprehensive empirical and analytical evidence of bias in speaker verification systems and their development processes. Furthermore, the research has presented new empirical evidence and insights on sources of bias in on-device ML that had previously not been observed.

Communication of Research Results

The final activity of the Design Science Research Method and a critical component of design science research is the communication of research findings to technical and non-technical audiences. I have published the results of the research on an ongoing basis in academic conferences and journals. In addition to the academic publications which have been referenced throughout this thesis and are included in the List of Publications, I have engaged extensively with academic, industry and civil society groups to communicate my research findings.

7.3. Findings from this Study

This section highlights the key findings for each goal and its supporting research question. The findings are presented sequentially, in the order in which the questions were posed. Goal 3 and RQ3 are the outcome of the final design iteration, and mark the response to the overarching goal of creating design patterns for detecting and mitigating bias in the development of Edge AI.

7.3.1. Identifying Patterns

Chapter 4 launched the iterative design science research cycle, and addressed the question "which established approaches for bias detection and mitigation in ML can help re-

searchers and practitioners from other domains detect and mitigate bias in ML"? This chapter thus answered RQ1 and accomplished Goal 1. To support the pattern identification process, Section 4.3 proposed a conceptual model of high level categories that classifies patterns based on their scope and purpose. Pattern purposes are either analysis or design. The pattern scopes are dataset, evaluation, interaction, pre-, in- and post-processing. This classification corresponds with stages in the ML development workflow.

Design Patterns for Detecting and Mitigating Bias

The proposed conceptual model was used in Section 4.4 to identify patterns from existing software tools that have been developed to evaluate bias in ML systems. Software tools that have been developed for bias detection and mitigation present a good starting point for identifying design patterns that are used in practice. 5 tools were selected and reverse engineered to identify 23 patterns that had been implemented in the tools or described in the tool documentation. Below, the patterns that were identified in the tools, and 4 additional patterns that were identified in later chapters, are grouped by pattern scope and listed in alphabetical order.

Dataset Patterns

1. *Data quality distribution across groups (new, Chapter 5)*
2. Data quantity distribution across groups
3. *Data source (new, Chapter 5)*
4. Ground-truth label distribution across groups
5. Metadata approximation

Evaluation Patterns

6. Continuous testing
7. Disaggregated base metrics
8. Error tolerance
9. Evaluation datasets
10. Fairness assumptions
11. Group design
12. Group-based bias measure
13. Individual bias-measure
14. Meta-measure
15. Types of harm

Interaction Patterns

16. Visualising Metrics and Measures
17. *Data-driven design decisions (new, Chapter 6)*

Pre-processing Patterns

18. Decorrelating sensitive features
19. Feature editing

20. Sampling and reweighing

In-processing Patterns

21. Adversarial bias mitigation
22. Imposing fairness constraints on the optimisation objective
23. In-processing model selection

Post-processing Patterns

24. *Bias-aware model selection (new, Chapter 6)*
25. Output label redistribution
26. Threshold optimisation
27. Threshold selection

Using Patterns in Recipes

A pattern is intended to be used together with other patterns in a collection to achieve a particular objective. To this end, two pattern recipes were proposed in Section 4.5 to guide bias detection and mitigation processes. The patterns can be used in proactive or retrospective mode. In proactive mode, a recipe functions as design tool, for example to guide the process of creating a benchmark dataset or of measuring bias. In retrospective mode a recipe can be used to evaluate the quality of an existing process. The *Bias Measurement* and *Benchmark Dataset* recipes that were proposed are listed below.

Bias Measurement Recipe

- o 1 Types of harm
- o 2 Fairness assumptions
- + 3 Group design
- o 4 Metadata approximation
- + 5 Disaggregated base metrics
- o 6 Group-based bias measure
  - o Visualising metrics and measures
  - o Meta-measure

Benchmark Dataset Recipe

- + 1 Evaluation datasets
- + 2 Group design
- o 3 Metadata approximation
  - + Ground-truth label distribution across groups
  - + Data quantity distribution across groups
  - + Data quality distribution across groups
  - + Data source

Validating Patterns against Requirements

Chapter 4 set out 17 requirements for validating the patterns. The requirements were grouped by pattern form, function, content and interaction. Requirements 1 to 4, 10 and 12 to 17 which address pattern form, content and interaction, were validated in Chapter 4. The functional requirements 6 to 9 and content requirement 11 were validated in the following chapters. Validating requirement 5 remains an area of future work.

7.3.2. Validating Pattern Utility

In Chapter 5 the patterns underwent two develop-demonstrate-validate design iterations to address the question "to what extent are the design patterns elicited in RQ1 useful for detecting and mitigating bias in ML"? The chapter validated the utility of the patterns for detecting and mitigating bias in a speaker verification use case, thus answering RQ2 and accomplishing Goal 2.

The use case has high societal relevance and a strong connection to Edge AI, as speaker recognition systems are widely deployed in voice-based systems in call centres and on devices. On the one hand they contribute an important component to trustworthy Edge AI by securing voice-enabled services that are activated by invoking a device. On the other hand they pose a privacy threat by enabling a pervasive, hidden and distributed surveillance infrastructure in people's most personal environments. Prior to the research that supported this thesis, only limited research had investigated bias in speaker recognition.

Using Design Patterns to Evaluate Bias

In Section 5.3 the *Benchmark Dataset* and *Bias Measurement* recipes were used to detect bias in the VoxCeleb speaker recognition challenge. The section is based on research published in [128]. The recipes were validated in retrospective mode. The *Benchmark Dataset* recipe was used to conduct an analysis of the VoxCeleb1 benchmark dataset, and the *Bias Measurement* recipe was used to evaluate bias in baseline models trained on VoxCeleb2. In the course of the study one new pattern was identified. The *Data source* pattern is a dataset pattern with an analysis purpose that is included in the *Benchmark Dataset* recipe as a required, unordered pattern. The patterns in both recipes were used in the order specified by the recipe notation. Where applicable, each pattern analysis was preceded by a summary of the sections discussed in the pattern. The sections created an emergent structure that is helpful when using the patterns. Over time, they may be formalised into templates to follow during analysis.

The recipes enabled a robust and reproducible bias evaluation that helped to surface sources of bias in the VoxCeleb speaker recognition challenge, and directly supported the conclusion of the study that bias exists at every stage in the development workflow of speaker verification systems. The study thus successfully validated the recipes and patterns in the use case. Their general form makes them strong candidates for supporting bias detection in applications beyond the speaker recognition domain.

Using Design Patterns to Mitigate Bias Evaluation Pitfalls

While validating the pattern recipes for bias detection in speaker recognition, it became evident that poorly executed bias evaluations have shortcomings that compromise their validity. The recipes can help to overcome these shortcomings, thus mitigat-

ing pitfalls in bias evaluations. Sections 5.4 and 5.5 examined shortcomings in benchmark datasets and bias measurement respectively.

Section 5.4 investigated the impact of the *Evaluation datasets* and *Data quantity distribution across groups* patterns on speaker verification evaluation outcomes by studying how utterance pairings influence the representativeness and sufficiency of speaker groups in benchmark datasets. The section is based on research published in [130]. The study found that the difficulty of utterance pairs impacts the outcome of speaker verification evaluations, and that randomised utterance pairings that neglect to account for the difficulty of pairs can result in significant performance variation if the utterance pair count per speaker is low. To overcome these pitfalls, the study proposed an algorithm for generating speaker verification evaluation datasets that accounts for pair difficulty, representativeness and sufficiency of speaker representation. The study showed empirically that datasets generated with this algorithm provide a more robust evaluation across people with different nationalities and genders. As a result, the section introduced a new dataset pattern for analysis purposes, the *Data quality distribution across groups* pattern. This new pattern has been included in the *Benchmark Dataset* recipe as a required, unordered pattern.

Section 5.5 compared approaches to measuring bias in speaker verification to investigate how fairness assumptions, base metrics, bias measures and meta-measures influence the evaluation outcome. The study found that bias evaluations are highly sensitive to the base metrics that are being compared. Ratio-based bias measures were found to offer better interpretability, and were more sensitive to base metrics of a small magnitude. One of the two meta-measures compared was shown to be sensitive to the order of magnitude of the base metric and the weights assigned to false positive and false negative error rates. The study empirically demonstrated two important considerations for bias evaluations. Firstly, using the *Group-based bias measure* and *Meta-measure* patterns without carefully considering the application context can lead to contradictory bias claims and erroneous conclusions. Secondly, even though the *Types of harm* and *Fairness assumptions* patterns are optional in the *Bias Measurement* recipe, they contextualise bias evaluations and provide the necessary background to choose between base metrics, bias measures and meta-measures.

Together, these two studies demonstrated the utility of the patterns and recipes for identifying and subsequently mitigating pitfalls in bias evaluations. The studies thus provide a promising affirmation of the effectiveness of the design patterns for mitigating bias in ML.

7.3.3. Extending Patterns to Edge AI

The final develop-demonstrate-validate design iteration was presented in Chapter 6, which concluded the design science research cycle. The chapter addressed the question "which aspects of Edge AI may affect bias"? In doing this, the chapter extended the design patterns to an on-device audio keyword spotting (KWS) use case, and adapted

them to incorporate new knowledge specific to detecting or mitigating bias in Edge AI systems. It thus answered RQ3 and accomplished Goal 3.

From ML to Edge AI, RQ3 investigated sources of bias in Edge AI that are not studied in the algorithmic fairness literature. On-device ML comprises Edge AI topologies where ML training or inference happens on resource constrained end devices. While on-device ML applications are pervasive, contextual and highly localised, they must also confront limited memory, computing capabilities, and energy resources. These constraints pose trade-offs that are known to affect prediction accuracy. However, no prior research had investigated how resource constrained on-device ML settings affect bias. Where the previous chapter validated the *Benchmark Dataset* and *Bias Mitigation* recipes for detecting and mitigating bias in pre-trained ML models, this chapter validated them for detecting and mitigating bias during on-device ML model development. Furthermore, the chapter extended the patterns to address sources of bias that arise in this setting. The chapter is based on prior research published in [131]. The findings of this study and their implications for the thesis are discussed next.

Aspects of On-device Keyword Spotting that Impact Bias

Section 6.2 described how design choices in on-device ML impact KWS systems and how engineers navigate these choices in the development workflow. The design choices under investigation pertained to the data sample rate, pre-processing parameters and input features, the model architecture, and pruning hyper-parameters. The first three choices arise during model training. Pruning is a post-processing operation and the choice of pruning hyper-parameters arises during model optimisation. In Section 6.3 the pattern recipes were used in proactive mode to guide the experiment design. First, the *Benchmark Dataset* recipe was used to construct KWS datasets for training and evaluation. Next, the *Bias Measurement* recipe was used to set up the bias evaluation in a rigorous and reproducible manner. Section 6.4 presented the results and analysis of the empirical study. The findings on sources of bias in design choices of development workflows in on-device KWS systems are summarised below.

- Model accuracy is lower at lower sample rates and for light-weight architectures. The median and interquartile range of bias tend to be greater at lower sample rates and for light-weight architectures. The direction of bias is strongly influenced by the training dataset. Overall, male speakers are favoured by models for all but one dataset.
- Feature type and dimensions impact KWS accuracy and bias. Their effect is further influenced by the training dataset. In general, MFCC type features perform better than log Mel spectrograms. However, they can also increase bias, prejudicing models against females and favouring males. For MFCC features, fewer dimensions (i.e. cepstral coefficients and Mel filter banks) can reduce computational demands with a negligible impact on accuracy and bias.

- Polynomial decay is a more robust pruning schedule than constant sparsity, and a larger pruning learning rate, like 0.001, reduces the likelihood of unintended bias and unexpected accuracy degradation. These design choices are particularly important when pruning models to sparsities greater than 50%, beyond which accuracy and bias can deteriorate dramatically. The increase in bias due to pruning is greater for smaller architectures and at lower sample rates. This trend is stronger at smaller learning rates. Training, validation and test datasets need to be large enough and representative across groups of users to ensure robust results and avoid bias.

These findings present new insights on bias in on-device ML. They also show that the recipes can be used successfully to set up and conduct a bias evaluation in the Edge AI domain.

Adapting Design Patterns to On-device ML

Based on the results analysis and findings, Section 6.5 proposed two new patterns that present low effort strategies for mitigating bias in on-device ML. Both patterns have a design purpose and give engineers an active role in detecting and mitigating sources of bias in on-device ML development. The *Bias aware model selection* pattern is a post-processing pattern that engineers should use to ensure that models selected after training and pruning have high accuracy and low bias. The *Data-driven design decisions* pattern is an interaction pattern that engineers should use to support their design decisions during on-device ML development with targeted and systematic experimentation. The recipes remained unaltered.

This chapter has demonstrated that the patterns can be adapted to include new design knowledge specific to the Edge AI setting and that the recipes extend to this application area. This marks the end of the iterative design cycles and concludes the design science research process. In the next section I highlight the contributions of the thesis.

7.4. Limitations

The design patterns that this thesis proposes are a novel approach to formalising a generalisable solution to the socio-technical problem of detecting and mitigating bias in Edge AI. In this section I discuss the limitations of the research process and outputs. The limitations stem from design patterns being a work in progress and from the voice-activation focus of the use cases. I also deliberate on the limits that bias detection and mitigation hold for building fair AI, and on aspects of socio-technical design beyond technical development processes.

7.4.1. Design Patterns as a Work in Progress

This dissertation has followed an iterative design process to identify 27 design patterns for detecting and mitigating bias in Edge AI. The patterns are grounded in best-practice from ML fairness and have been evaluated in use cases for voice-activation applications. Despite the rigour pursued in their development, the design patterns are a work in progress. They should be regarded as a novel and promising approach to address a pressing challenge in the development of trustworthy AI, not as an immutable and complete set of patterns.

This limitation is justified for two reasons: on the one hand, given the diversity of applications that use AI, it is unlikely that a complete set of design patterns for detecting and mitigating bias will be identified in the short term. As new application domains adopt AI, so new sources of bias may emerge that lead to yet unidentified patterns. On the other hand, detecting and mitigating bias in AI is itself an emerging area of research, and knowledge in this field is continuously evolving. Even if it were possible to capture a complete set of design patterns, best-practice approaches will continue to emerge as the field matures. This new knowledge ought to be incorporated into the design patterns. Rather than judging the success of the design patterns by their completeness, I will consider the research objective fulfilled if future studies revisit and revise the design patterns to iteratively improve our understanding of how to build trustworthy Edge AI.

7.4.2. Generalisability beyond Use Cases

The use cases for evaluating and improving the design patterns are limited to two voice-activation tasks: speaker recognition and keyword spotting (KWS). The choice to focus on voice-activation was done carefully: firstly, voice-based services are a dominant application of Edge AI in today's Internet of Things. Secondly, voice-activation grants access to downstream services, which means that failures lead to service denial, security and privacy risks. This makes bias particularly problematic, as it can disadvantage groups of people in their interactions with the world. Thirdly, in relation to other AI-enabled components, voice-activation tasks can be investigated with minimal resources, which makes them a logical first step for investigating bias in Edge AI.

It is to be expected that the choice of use cases constrained the direction in which the patterns evolved during the iterative design process. A particular constraint is that the use cases only consider AI tasks where disparate error rates constitute bias. The patterns thus do not consider other types of bias, such as representation bias which is common in image and language processing tasks [178]. The recipes further influenced which patterns were validated. Nonetheless, I expect that the proposed patterns have a high degree of transferability to new applications. For example, within the voice-activation use cases, we investigated on-device KWS, but speaker recognition on the cloud. While this setup resembles current architectures of voice assistants, speaker

recognition and KWS are becoming more tightly coupled, and speaker recognition is likely to move from the cloud to devices in the coming years. Thus, the post-processing patterns that were identified for on-device KWS are likely to be effective for future speaker recognition systems. Similarly, the dataset patterns identified in the speaker recognition use case will also be relevant to KWS.

In Section 7.5 I discuss recommendations and future work to evaluate the design patterns for different speech processing tasks, and to consider them for new data modalities such as image, video and motion tracking data. While pattern instantantiations specific to these modalities may be necessary, I hypothesise that the recipes, and the conceptual model that categorises patterns by scope and purpose will remain relevant.

7.4.3. The Long Road from Bias to Fairness

Artifacts like the design patterns can play an important role in reducing discrimination in AI-enabled technologies. Yet, even though the design patterns seek to provide a generalisable solution to recurring problems in bias detection and mitigation, fairness cannot be generalised. Consequently, tools for detecting and mitigating bias cannot guarantee that Edge AI will be fair.

Fairness is highly context specific, and definitions of fairness vary across stakeholders and applications. By measuring disparities in error rates between groups of people, as I do in this research, it is possible to reduce the complex socio-technical phenomenon of bias to a measurable quantity that can then be controlled and managed. However, even if it were to be possible to create an unbiased application, it may still not meet the requirements that would make it fair. For example, consider a smartphone with a KWS system that listens for particular words to support the creation of a consumer profile for advertising purposes. Even if this system runs locally on device and performs equally well for all people (i.e. it is unbiased), it may not be fair. Firstly users may not have opted in to their voice data being used in this way. Then, even if they did consent, their ability to appreciate the consequences of this choice will differ across individuals and may change over time. More so, the consequences themselves may change over time. The impact of the advertising recommendations may also impact people disparately. They could, for example, result in greater harm when making recommendations to children and teenagers than to adults.

The design patterns that this thesis proposes only partially support the difficult transition from detecting and mitigating bias to building fair Edge AI. While the *Types of harm* and *Fairness assumptions* patterns bring fairness to the forefront, they are not a substitute for a deeper inquiry into values and fairness. Rather than dwelling on this limitation, I see it as an opportunity to invite further exploration of the rich suite of techniques made available through other design practices. I envision that the design patterns will be one set of tools alongside other approaches such as responsible, value-sensitive, human-centred and participatory design. Together with such patterns and practices, additional approaches like building diverse teams and fostering a team cul-

ture in which people feel safe and appreciated will be necessary for moving from bias detection and mitigation to fair Edge AI.

7.4.4. Multiple Facets of Socio-Technical Design

In this dissertation I position the design patterns as a tool for detecting and mitigating bias in the technical development process of Edge AI systems. Technical development activities, while essential, are not the only design processes where decisions are made in the development and deployment of Edge AI. When considering the broader socio-technical context of Edge AI, factors like management practices, interpersonal relationships, cultural norms, organisational rules and so forth strongly influence the nature of the technology that is developed. This research is limited in that it focuses only on technical design, and does not consider institutional and process design [156]. Consequently, the research has limited its scope to bias in technological artifacts, excluding cognitive, systemic and other forms of bias. Despite the scope being intentionally constrained in this regard, considering institutional and process design alongside the design patterns is a necessary extension of the research topic.

7.5. Recommendations

In this section I consider recommendations and future work pertaining to the design patterns. In brief, I recommend future work that focuses on the adoption, the adaptation and the instantiation of the design patterns. While not covered specifically in these recommendations, the use case evaluations in this thesis have highlighted that mitigating bias in speech processing technologies and on-device machine learning are, in their own right, important areas for future work.

7.5.1. Adoption of Design Patterns

The design patterns have been developed from existing software tools that detect and mitigate bias, and through empirical and analytical studies. They are thus grounded in state-of-the art knowledge and in rigorous scientific practice. Ultimately, my motivation for studying design patterns was however a pragmatic one: I want to contribute towards a future in which Edge AI is non-discriminatory, fair and aligned with societal values. To have practical relevance, the design patterns thus need to be adopted by engineers and developers designing Edge AI products. The next step in the development of the design patterns is then to study their use and adoption in practice. The utility of the design patterns is dependent firstly on them being useful to engineers and secondly on their effectiveness at detecting and mitigating bias during product development. Research questions that may inform future work are:

- What factors influence engineers to adopt practices to detect and mitigate bias,

and by extension the design patterns?

- What support is necessary to encourage and enable engineers to adopt the design patterns?
- How should the design patterns be integrated into current Edge AI development workflows?
- Does the adoption of design patterns for detecting and mitigating bias in Edge AI development lead to more inclusive and less biased Edge AI products and services?

7.5.2. Adaptation to New Use Cases and Domains

To evaluate the effectiveness of the design patterns more broadly, an important area of future work is to observe if their uptake assists the advancement of work on bias in new domains. Alongside adoption, adaptation of the design patterns to new use cases is thus important to ensure their utility. Bias in speech processing technologies is currently an underexplored research area. The design patterns naturally lend themselves to use cases for different speech processing tasks, like automatic speech recognition, language and accent detection, and emotion recognition from speech. Of particular interest is the transfer of knowledge on bias mitigation interventions between related tasks. Beyond speech processing, new data modalities that constitute popular Edge AI applications, for example computer vision and activity detection tasks, as well as on-device ML in the health and well-being domains, are promising frontiers for applying the design patterns. Research questions that may inform future work in this area are:

- To what extent are the design patterns useful for detecting and mitigating bias in new data modalities?
- How can the design patterns be extended to types of bias that are not easily quantified as error rate disparities?
- To what extent can bias mitigation interventions between related tasks be transferred?
- What are enabling factors that support the extension of the design patterns to new domains?

7.5.3. Instantiation in Software Tools

In this thesis I identified the initial design patterns from software tools that are used to detect and mitigate bias in ML. These tools were primarily developed to investigate bias in decision-making systems and thus do not apply directly to the Edge AI setting. A practical extension of the design patterns is thus to instantiate them in new software tools for detecting and mitigating bias in Edge AI. The bt4vt[2] python library, which

[2] https://github.com/wiebket/bt4vt

diagnoses bias in speech processing tasks, is a software tool that explicitly instantiates several of the design patterns. Instantiating the design patterns as software tools can support their adoption and adaptation. Relevant research questions that may inform future work are:

- What software tools are needed to make the design patterns accessible to design teams?
- What are the requirements for instantiating the design patterns in a software library?
- What functionality does a software library need to provide to support the adoption and adaptation of the design patterns?
- How should such a software library be designed to instantiate the design patterns and make them auditable?

8

Epilogue

Reflection

For those things we call AI are adapted to human goals and purposes. They are what they are in order to satisfy our needs and desires.

"Under what circumstances is AI good?" Does good mean something important and vital that goes, ultimately, to the nature of human life? Is making AI better the same as making it good? If better means merely efficient, then it is not.

AI changes our aims which are embodied in AI. When developing AI, is searching for something that helps human life a formal part of what we are searching for? Or are we primarily searching for – what should I call it – good technical performance? This seems to me a very, very vital issue.

Human generated remix of H. Simon and C. Alexander's intellectual property.

The development of AI, its speculated benefits and anticipated risks, are not inevitable and predetermined. Rather, like many technologies that humans have created before, the advancement of AI exists within and is funded by governments, corporations and organisations that are collections of people, acting strategically and politically to advance their interests. *People*, be they developers or users, managers or funders, citizens or policy makers, play a central role in aligning the development of AI with societal goals. However, aligning any technology development with societal objectives is a complex and difficult undertaking; not one amenable to brute-force algorithmic optimisation.

Technology does not reflect the world as is, but as we create it to be. It is our duty to pause and reflect as we continue to develop AI. As with all technologies, people will need to choose between trade-offs when creating AI. The choices that are made will reinforce certain values and ignore others, amplify certain people while leaving others unaccounted for. Who gets to choose? Who is accounted for? Whose goals and purposes are artifacts adapted to? Whose needs and desires does AI satisfy? These are very, very vital issues if we are to develop trustworthy AI.

A design perspective on AI risks and harms

Over the past decade the momentum of AI research has been heavily weighted towards mathematics, probability theory, statistical learning algorithms and computing systems; areas of science that specialise in quantifying and quantitatively modelling the world. Today, scientific advances in these domains have resulted in breakthrough technologies that have leaped almost overnight from the lab into products used by billions of people. Text generation and music creation, article summarisation and translation, identity cloning and image manipulation are now accessible to anybody with the desire to use *or* abuse these technologies. With the leap from the lab to the world, the future of AI has moved from observation to creation, from science to technology.

Still, the dominant narrative of AI remains one where normative choices are framed as absolutes, where the existence of trade-offs is ignored or explained away, and where "good technical performance" is the yardstick by which AI is measured. The rapid development of AI is cast as inevitable, justified by the power that the control of AI will bestow upon its guardians. We release new AI onto the world at breakneck speed, having learnt little from the biodiversity, sustainability and climate crises that industrialisation and the fossil fuel era have brought upon us, let alone from the "fail fast and break things" mentality of the Internet and social media boom. As it was then, so it is now – 8 billion people are a sandbox for the potential of a profitable "killer app", no matter the future cost to mental, physical, environmental and societal health.

At this inflection point it is necessary to pause and contemplate the world that we, as technologists, are creating. However, when pausing to do that, we quickly run into a conceptual and a tooling deficit, as AI research at large has not been concerned with the normative. By extension, the development of AI systems has been abstracted from the choices, perspectives, values and ambitions of the individuals and institutions that create, manage and use them. While research about the risks and responsible design of AI technologies is rapidly gaining traction, at a moment in time where guidance on the confluence of AI and human society is most needed, we find the factions of researchers working at the forefronts of AI ethics and AI safety engaged in ideological battles about subjective beliefs on short-term and long-term risks. These battles do not serve society, and the dualistic framing of an AI future that is either rigidly regulated or libertarian leaves no room for finding common ground.

Design and systems thinking for trustworthy AI

In light of this, it is perhaps surprising that several years of timely funding of trustworthy AI research have yielded only limited practical outputs that make the realisation of trustworthy AI tractable at this moment in time. We still seem to have more questions than answers, together with plenty new algorithms for privacy, explainability and fairness, for security and robustness. More quantification, but neither greater understanding, nor better tools for envisioning and interrogating systems beyond a narrowly defined performance metric. Attention is all you need[1] to build modern AI systems, echos the chorus of AI researchers; and self-anointed, well-intentioned technologists will ensure their ethical and safe deployment. In reality, intention is insufficient for any technology to ensure that it supports societal values and well-being. In my twenties, as CEO of Engineers Without Borders South Africa, I observed hundreds of well-intentioned development projects fail. Rarely was this failure attributable to our inability to conceive and engineer complex technologies. Almost always could it be attributed to an incomplete understanding of the changing project context over its lifetime, to the omission of critical (often non-technical) system components that were hidden to the engineers when the project was conceptualised, and to a lack of under-

[1] A reference to the famous research paper that first introduced Transformer models

standing of how social and institutional system components determined technology use and evolution.

A reductionist paradigm that examines and improves only the technical components of AI systems is not enough to overcome these challenges. Instead, a systems perspective that considers relationships, interactions, forces, incentives, and system boundaries in an integral manner is urgently needed. In the disciplines of socio-technical design, design thinking and systems thinking, skills and approaches for modelling and tackling complex systems problems have been taught for several decades. As systems thinker Donella Meadows eloquently stated, "there are no separate systems. The world is a continuum." Boundaries are an abstraction that we invent to articulate a problem more clearly, to stay sane. If boundaries are too narrow, system behaviour will surprise us. If boundaries are too wide, systems become too complex to analyse and may obscure the solutions we seek, rather than reveal them.

Expanding trustworthy AI system boundaries to the socio-technical and practical

In having moved from the lab to the world, the narrow boundaries drawn around technical AI components only offer a limited view on the technology. From this vantage point, risks and harms, today and in future, cannot be anticipated adequately. If AI technology is to be trusted to support our collective existence and promote social integrity and cohesion, the narrow boundaries of AI systems must be expanded. With this view, I lay down my own cognitive bias that has shaped the trajectory of this PhD thesis. When it comes to trustworthy AI, research needs to extend beyond the technical into the socio-technical, beyond the theoretical into the practical. Omitting to do this may leave us with methods for detecting technical bias and engineers that want to test for it, but without tools and processes for doing this. Or with regulation that demands AI audits, but no budgets to oversee them, no technical know-how to implement them and no human resources to conduct them.

The design patterns developed in this thesis and the use cases in which they have been applied demonstrate the contribution that practical artifacts and interventions developed through a rigorous design science research approach can make to the development of trustworthy AI. I speculate that the design patterns can serve multiple functions: they are a tool for robust product development, a framework for interdisciplinary communication, and perhaps a set of concepts from which AI audits can be reverse engineered. However, today this type of research still requires a pioneering attitude, as it is scarce to use design methods in trustworthy AI research. My expectation is that this will change in the near future, as the rapid development of AI applications will challenge us to adopt new approaches to assure their trustworthy deployment in a consistent and clear manner. Returning to Donella Meadows, in systems "at any given time, the input that is most important ... is the one that is most limiting". At this moment in time, I audaciously call out that a paradigm shift from a technology-first mindset to a systems-thinking and socio-technical design perspective is the limiting factor in realising trustworthy AI.

Appendix

Statistical Analysis

Design Choices Arising During Model Training

Effects of model training design choices (initial)

```
model_inital = 'metric ~ C(dataset_name, Sum)+C(model_arch,
    Sum)C(resample_rate, Sum)+C(mfccs, Sum)+C(mel_bins,
    Sum)+C(frame_length, Sum)+C(frame_step, Sum)+C(window_fn,
    Sum)+C(dataset_name, Sum)*C(model_arch, Sum)*C(resample_rate,
    Sum)*C(mfccs, Sum)*C(mel_bins, Sum)+C(dataset_name, Sum)*C(model_arch,
    Sum)*C(resample_rate, Sum)*C(frame_length, Sum)*C(frame_step,
    Sum)*C(window_fn, Sum)'
```

Model 1: First factorial ANOVA interaction model for model training design choices

Effects of model training design choices on MCC

```
model_final_mcc = 'mcc ~ C(dataset_name, Sum)+C(model_arch,
    Sum)+C(resample_rate, Sum)+C(mfccs, Sum)+C(mel_bins,
    Sum)+C(dataset_name, Sum)*C(resample_rate, Sum)*C(mfccs,
    Sum)+C(model_arch, Sum)*C(mfccs, Sum)*C(mel_bins, Sum)+C(dataset_name,
    Sum)*C(model_arch, Sum)*C(mfccs, Sum)+C(frame_length,
    Sum)+C(frame_step, Sum)+C(model_arch, Sum)*C(frame_length,
    Sum)*C(frame_step, Sum)'
```

Model 2: Final factorial ANOVA interaction model for effect of model training design choices on MCC

Effects of model training design choices on reliability bias

```
model_final_bias = 'reliability_bias ~ C(dataset_name, Sum)+C(model_arch,
    Sum)+C(resample_rate, Sum)+C(dataset_name, Sum)*C(resample_rate,
    Sum)+C(mfccs, Sum)+C(mel_bins, Sum)+C(dataset_name, Sum)*C(model_arch,
    Sum)*C(mfccs, Sum)+C(dataset_name, Sum)*C(mel_bins,
    Sum)+C(frame_length, Sum)'
```

Model 3: Final factorial ANOVA interaction model for effect of model training design choices on reliability bias

Design Choices Arising During Model Optimization

Effects of pruning hyper-parameter design choices (initial)

```
model_inital = 'delta_metric ~ mcc_baseline + reliability_bias_baseline +
    C(dataset_name, Sum)+C(model_arch, Sum)+C(resample_rate,
    Sum)+C(pruning_learning_rate, Sum)+C(pruning_schedule,
    Sum)+C(pruning_frequency, Sum)+C(pruning_final_sparsity,
    Sum)+C(dataset_name, Sum)*C(model_arch, Sum)*C(resample_rate,
    Sum)*C(pruning_learning_rate, Sum)*C(pruning_schedule,
    Sum)*C(pruning_frequency, Sum)*C(pruning_final_sparsity, Sum)'
```

Model 4: First factorial ANOVA interaction model for pruning hyper-parameters

Effects of pruning hyper-parameter design choices on change in MCC

```
model_final_delta_mcc = 'delta_mcc ~ mcc_baseline +
    reliability_bias_baseline + C(dataset_name, Sum)+C(model_arch,
    Sum)+C(resample_rate, Sum)+C(pruning_learning_rate,
    Sum)+C(pruning_schedule, Sum)+C(pruning_frequency,
    Sum)+C(pruning_final_sparsity, Sum)+C(model_arch,
    Sum)*C(pruning_learning_rate, Sum)*C(pruning_schedule,
    Sum)*C(pruning_final_sparsity, Sum)+C(dataset_name, Sum)*C(model_arch,
    Sum)*C(resample_rate, Sum)*C(pruning_final_sparsity,
    Sum)+C(dataset_name, Sum)*C(pruning_learning_rate,
    Sum)*C(pruning_schedule, Sum)+C(dataset_name, Sum)*C(pruning_schedule,
    Sum)*C(pruning_final_sparsity, Sum)+C(dataset_name,
    Sum)*C(pruning_learning_rate, Sum)*C(pruning_final_sparsity,
    Sum)+C(resample_rate, Sum)*C(pruning_learning_rate,
    Sum)*C(pruning_final_sparsity, Sum)'
```

Model 5: Final factorial ANOVA interaction model for effect of pruning design choices on change in MCC

Effects of pruning hyper-parameter design choices on change in reliability bias

```
model_final_delta_bias = 'delta_reliability_bias ~
    mcc_baseline+reliability_bias_baseline+C(dataset_name,
    Sum)+C(model_arch, Sum)+C(resample_rate, Sum)+C(pruning_learning_rate,
    Sum)+C(pruning_schedule, Sum)+C(pruning_frequency,
    Sum)+C(pruning_final_sparsity, Sum)+C(dataset_name, Sum)*C(model_arch,
    Sum)*C(resample_rate, Sum)*C(pruning_learning_rate,
    Sum)+C(dataset_name, Sum)*C(pruning_learning_rate,
    Sum)*C(pruning_final_sparsity, Sum)+C(pruning_schedule,
    Sum)*C(pruning_final_sparsity, Sum)+C(model_arch,
    Sum)*C(pruning_final_sparsity, Sum)'
```

Model 6: Final factorial ANOVA interaction model for effect of pruning design choices on change in reliability bias

List of Figures

List of Tables

References

[1] ACM FAccT Executive Committee. ACM FAccT Strategic Plan, 2020. URL https://facctconference.org/static/docs/strategic_plan.pdf.

[2] M. Adda-Decker and L. Lamel. Do speech recognizers prefer female speakers? In *Proc. Interspeech*, pages 2205–2208, 2005. doi: 10.21437/Interspeech.2005-699.

[3] C. Alexander. *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, Massachusetts, 7th edition, 1973.

[4] C. Alexander. The Origins of Pattern Theory, the Future of the Theory, And the Generation of a Living World, 1996. URL https://www.patternlanguage.com/archive/ieee.html.

[5] C. Alexander. *The Nature of Order: The process of creating life*. Center for Environmental Structure series. Center for Environmental Structure, 2002. ISBN 9780972652926.

[6] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, USA, 1977. ISBN 0195019199. URL https://www.patternlanguage.com/bookstore/pattern-language.html.

[7] R. Alvarez and H. J. Park. End-to-end streaming keyword spotting. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6336–6340, 2019. ISBN 9781538646588.

[8] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann. Software Engineering for Machine Learning: A Case Study. *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, pages 291–300, 2019. doi: 10.1109/ICSE-SEIP.2019.00042.

[9] A. Armoush. *Design Patterns for Safety-Critical Embedded Systems*. PhD thesis, RWTH Aachen University, 2010.

[10] R. Baeza-Yates. Bias on the web. *Communications of the ACM*, 61(6):54–61, 2018. ISSN 15577317. doi: 10.1145/3209581.

[11] Z. Bai and X. L. Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140:65–99, 2021. ISSN 18792782. doi: 10.1016/j.neunet.2021.03.004.

[12] A. Balayn and S. Gürses. Beyond Debiasing: Regulating AI and its inequalities. Technical report, European Digital Rights, 2021. URL https://edri.org/our-work/if-ai-is-the-problem-is-debiasing-the-solution/.

[13] C. Banbury, V. J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Kiraly, P. Montino, D. Kanter, S. Ahmed, D. Pau, U. Thakker, A. Torrini, P. Warden, J. Cordaro, G. Di Guglielmo, J. Duarte, S. Gibellini, V. Parekh, H. Tran, N. Tran, N. Wenxu, and X. Xuesong. MLPerf Tiny Benchmark. In *Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, 2021. URL http://arxiv.org/abs/2106.07597.

[14] C. R. Banbury, V. J. Reddi, M. Lam, W. Fu, A. Fazel, J. Holleman, X. Huang, R. Hurtado, D. Kanter, A. Lokhmotov, D. Patterson, D. Pau, J.-s. Seo, J. Sieracki, U. Thakker, M. Verhelst, and P. Yadav. Benchmarking TinyML Systems: Challenges and Direction. 2020. URL http://arxiv.org/abs/2003.04821.

[15] H. Baniecki, W. Kretowicz, P. Piatyszek, J. Wisniewski, and P. Biecek. dalex: Responsible Machine Learning with Interactive Explainability and Fairness in Python. Technical report, 2021. URL http://jmlr.org/papers/v22/20-1473.html.

[16] S. Barocas, M. Hardt, and A. Narayanan. Fairness and Machine Learning: Limitation and Oppotunities. *Fairness and Machine Learning: Limitation and Oppotunities*, 2019. URL https://fairmlbook.org.

[17] G. Baxter and I. Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23(1):4–17, 2011. ISSN 09535438. doi: 10.1016/j.intcom.2010.07.003. URL http://dx.doi.org/10.1016/j.intcom.2010.07.003.

[18] R. K. Bellamy, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, Y. Zhang, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, and S. Mehta. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4-5), 2019. ISSN 21518556. doi: 10.1147/JRD.2019.2942287.

[19] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods and Research*, 50(1):3–44, 2021. ISSN 15528294. doi: 10.1177/0049124118782533.

[20] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods and Research*, 50(1):3–44, 2021. ISSN 15528294. doi: 10.1177/0049124118782533.

[21] D. Bermuth, A. Poeppel, and W. Reif. Jaco : An Offline Running Privacy-aware Voice Assistant. pages 618–622, 2022.

[22] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 9780387310732.

[23] P. R. Blake, K. McAuliffe, J. Corbit, T. C. Callaghan, O. Barry, A. Bowie, L. Kleutsch, K. L. Kramer, E. Ross, H. Vongsachang, R. Wrangham, and F. Warneken. The ontogeny of fairness in seven societies. *Nature*, 528(7581):258–261, 12 2015. ISSN 14764687. doi: 10.1038/nature15703.

[24] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach. Language (Technology) is Power: A Critical Survey of "Bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, 2020. doi: 10.18653/v1/2020.acl-main.485.

[25] T. Bolukbasi, K.-w. Chang, J. Zou, V. Saligrama, and A. Kalai. Man is to Computer Programmer as Woman is to Homemaker ? Debiasing Word Embeddings. In *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4356 – 4364, 2016.

[26] A. Bouguettaya, Q. Z. Sheng, B. Benatallah, A. G. Neiat, S. Mistry, A. Ghose, S. Nepal, and L. Yao. An internet of things service roadmap. *Communications of the ACM*, 64(9):86–95, 2021. ISSN 15577317. doi: 10.1145/3464960.

[27] A. Bower, S. N. Kitchen, L. Niss, M. J. Strauss, A. Vargas, and S. Venkatasubramanian. Fair Pipelines. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, number August, Halifax, Canada, 2017. URL http://arxiv.org/abs/1707.00391.

[28] M. Buchheit, W. Hickie, F. Hirsch, and S. Schrecker. AI Trustworthiness Challenges and Opportunities Related to IIoT. *IIC Journal of Innovation*, pages 1–14, 2019.

[29] J. Buolamwini and T. Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of Machine Learning Research: Conference on Fairness, Accountability, and Transparency*, volume 81, pages 1889–1896, 2018.

[30] J. Buolamwini and T. Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of Machine Learning Research: Conference on Fairness, Accountability, and Transparency*, volume 81, pages 1889–1896, 2018.

[31] H. Cai, C. Gan, L. Zhu, and S. Han. TinyTL: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 2020-Decem(NeurIPS), 2020. ISSN 10495258.

[32] C. Canel, T. Kim, G. Zhou, C. Li, H. Lim, D. G. Andersen, M. Kaminsky, and S. R. Dulloor. Scaling Video Analytics on Constrained Edge Nodes. In *Proceedings of the 2nd SysML Conference*, Palo Alto, US, 2019. doi: 10.1109/temc.2019.2945897.

[33] A. Castelnovo, R. Crupi, G. D. Gamba, G. Greco, A. Naseer, D. Regoli, and B. S. Miguel Gonzalez. BeFair: Addressing Fairness in the Banking Sector. In *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, pages 3652–3661. Institute of Electrical and Electronics Engineers Inc., 12 2020. ISBN 9781728162515. doi: 10.1109/BigData50022.2020.9377894.

[34] R. Chatila, V. Dignum, M. Fisher, F. Giannotti, K. Morik, S. Russell, and K. Yeung. Trustworthy AI. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12600 LNCS, pages 13–39. Springer Science and Business Media Deutschland GmbH, 2021. doi: 10.1007/978-3-030-69128-8{_}2.

[35] G. Chen, C. Parada, and G. Heigold. Small-Footprint Keyword Spotting Using Deep Neural Networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. IEEE, 2014. ISBN 9781467369978.

[36] J. Chen and X. Ran. Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*, 107(8): 1655–1674, 8 2019. doi: 10.1109/JPROC.2019.2921977.

[37] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Transactions on Information Systems*, 41(3):1–39, 7 2023. ISSN 1046-8188. doi: 10.1145/3564284.

[38] Y. Chen, B. Zheng, Z. Zhang, Q. Wang, C. Shen, and Q. Zhang. Deep Learning on Mobile and Embedded Devices: State-of-the-Art, Challenges, and Future Directions. *ACM Computing Surveys*, 53(4), August 2020. ISSN 0360-0300. doi: 10.1145/3398209.

[39] M. Chiang and T. Zhang. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, 2016. ISSN 23274662. doi: 10.1109/JIOT.2016.2584538.

[40] D. Chicco and G. Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 1 2020. ISSN 14712164. doi: 10.1186/s12864-019-6413-7.

[41] A. Chouldechova. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data*, 5(2):153–163, 2017. ISSN 2167647X. doi: 10.1089/big.2016.0047.

[42] J. S. Chung and A. Zisserman. Out of time: Automated lip sync in the wild. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10117 LNCS(i):251–263, 2017. ISSN 16113349. doi: 10.1007/978-3-319-54427-4{_}19.

[43] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B. J. Lee, and I. Han. In defence of metric learning for speaker recognition. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020-Octob:2977–2981, 2020. ISSN 19909772. doi: 10.21437/Interspeech.2020-1064.

[44] J. A. Colquitt, B. A. Scott, and J. A. LePine. Trust, Trustworthiness, and Trust Propensity: A Meta-Analytic Test of Their Unique Relationships With Risk Taking and Job Performance. *Journal of Applied Psychology*, 92(4):909–927, 7 2007. ISSN 00219010. doi: 10.1037/0021-9010.92.4.909.

[45] S. Corbett-Davies and S. Goel. The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. (Ec), 2018. URL http://arxiv.org/abs/1808.00023.

[46] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1296:797–806, 2017. doi: 10.1145/3097983.3098095.

[47] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. 5 2018. URL http://arxiv.org/abs/1805.10190.

[48] CPS Public Working Group. Framework for Cyber-Physical Systems : Volume 1 , Overview. Technical report, National Institute of Standards and Technology, 2017.

[49] J. Dastin. Amazon scraps secret AI recruiting tool that showed bias against women, 2018. URL https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

[50] K. Davis, E. Owusu, V. Bastani, L. Marcenaro, J. Hu, C. Regazzoni, and L. Feijs. Activity Recognition Based on Inertial Sensors for Ambient Assisted Living. *2016 19th International Conference on Information Fusion (FUSION)*, pages 371–378, 2016.

[51] M. J. Dawes and M. J. Ostwald. Christopher Alexander's A Pattern Language: analysing, mapping and classifying the critical response. *City, Territory and Architecture*, 4:17, 2017. doi: 10.1186/s40410-017-0073-1. URL https://doi.org/10.1186/s40410-017-0073-1.

[52] R. M. Dawes. The Robust Beauty of Improper Linear Models in Decision Making. Technical report, 1979.

[53] H. de Bruijn and P. M. Herder. System and actor perspectives on sociotechnical systems. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 39(5):981–992, 2009. ISSN 10834427. doi: 10.1109/TSMCA.2009.2025452.

[54] T. De Freitas Pereira and S. Marcel. Fairness in Biometrics: A Figure of Merit to Assess Biometric Verification Systems. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(1):19–29, 2022. ISSN 26376407. doi: 10.1109/TBIOM.2021.3102862.

[55] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *IEEE Internet of Things Journal*, 7(8):7457–7469, 2020. ISSN 23274662. doi: 10.1109/JIOT.2020.2984887.

[56] W. H. Deng, M. Nagireddy, M. S. A. Lee, J. Singh, Z. S. Wu, K. Holstein, and H. Zhu. Exploring How Machine Learning Practitioners (Try To) Use Fairness Toolkits. In *ACM Fairness, Accountability and Transparency 2022 (FAccT '22)*, pages 473–484. Association for Computing Machinery, 6 2022. ISBN 9781450393522. doi: 10.1145/3531146.3533113.

[57] P. J. Denning. Can Generative AI Bots Be Trusted? *Communications of the ACM*, 66(6):24–27, 6 2023. ISSN 0001-0782. doi: 10.1145/3592981. URL https://dl.acm.org/doi/10.1145/3592981.

[58] L. Devillers, F. Fogelman-Soulié, and R. Baeza-Yates. AI & Human Values: Inequalities, Biases, Fairness, Nudge, and Feedback Loops. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12600 LNCS, pages 76–89. Springer Science and Business Media Deutschland GmbH, 2021. doi: 10.1007/978-3-030-69128-8{_}6.

[59] S. Dey, A. Mukherjee, A. Pal, and P. Balamuralidhar. Embedded deep inference in practice: Case for model partitioning. *SenSys-ML 2019 - Proceedings of the 1st Workshop on Machine Learning on Edge in Sensor Systems, Part of SenSys 2019*, pages 25–30, 2019. doi: 10.1145/3362743.3362964.

[60] S. Dhar, J. Guo, J. Liu, S. Tripathi, U. Kurup, and M. Shah. On-Device Machine Learning: An Algorithms and Learning Theory Perspective. *ACM Transactions on Internet of Things*, 2(3), 2021. URL http://arxiv.org/abs/1911.00623.

[61] W. R. Dieter, S. Datta, and W. K. Kai. Power reduction by varying sampling rate. In *Proceedings of the 2005 international symposium on Low power electronics and design*, pages 227–232, 2005.

[62] A. Y. Ding, M. Janssen, and J. Crowcroft. Trustworthy and Sustainable Edge AI: A Research Agenda. In *Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 164–172, 2021. doi: 10.1109/TPSISA52974.2021.00019.

[63] A. Y. Ding, E. Peltonen, T. Meuser, A. Aral, C. Becker, S. Dustdar, T. Hiessl, D. Kranzlmüller, M. Liyanage, S. Maghsudi, N. Mohan, J. Ott, J. S. Rellermeyer 11, S. Schulte, H. Schulzrinne, G. Solmaz, S. Tarkoma, B. Varghese, L. Wolf, and A. Ding. Roadmap for Edge AI: A Dagstuhl Perspective. *ACM SIGCOMM Computer Communication Review*, 52(1):28–33, 2022. URL https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=21342.

[64] A. Y. Ding, E. Peltonen, T. Meuser, A. Aral, C. Becker, S. Dustdar, T. Hiessl, D. Kranzlmuller, M. Liyanage, S. Magshudi, N. Mohan, J. Ott, J. S. Rellermeyer, S. Schulte, H. Schulzrinne, G. Solmaz, S. Tarkoma, B. Varghese, and L. Wolf. Roadmap for Edge AI: A Dagstuhl Perspective. *ACM SIGCOMM Computer Communication Review*, 52(1):28 – 33, 2022. ISSN 19435819. doi: 10.1145/3523230.3523235. URL https://doi.org/10.1145/3523230.3523235.

[65] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55 (10):78–87, 2012. ISSN 00010782. doi: 10.1145/2347736.2347755.

[66] J. F. Dovidio, K. Kawakami, C. Johnson, B. Johnson, and A. Howard. On the Nature of Prejudice: Automatic and Controlled Processes. Technical report, 1997.

[67] I. E. Dror and P. A. F. Fraser-Mackenzie. Cognitive Biases in Human Perception, Judgment, and Decision Making: Bridging Theory and the Real World. In K. Rossmo, editor, *Criminal Investigation Failures*. Taylor & Francis, 2008. URL http://users.ecs.soton.ac.uk/id/biometrics.html.

[68] A. K. Dwivedi, A. Tirkey, and S. K. Rath. Software design pattern mining using classification-based techniques. *Frontiers of Computer Science*, 12(5):908–922, 10 2018. ISSN 20952236. doi: 10.1007/s11704-017-6424-y.

[69] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, V. Galanos, P. V. Ilavarasan, M. Janssen, P. Jones, A. K. Kar, H. Kizgin, B. Kronemann, B. Lal, B. Lucini, R. Medaglia, K. Le Meunier-FitzHugh, L. C. Le Meunier-FitzHugh, S. Misra, E. Mogaji, S. K. Sharma, J. B. Singh, V. Raghavan, R. Raman, N. P. Rana, S. Samothrakis, J. Spencer, K. Tamilmani, A. Tubadji, P. Walton, and M. D. Williams. Artificial Intelligence (AI): Multidisciplinary perspectives on

emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, 57, 4 2021. ISSN 02684012. doi: 10.1016/j.ijinfomgt.2019.08.002.

[70] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. *ITCS 2012 - Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012. doi: 10.1145/2090236.2090255.

[71] M. Dzambic, J. Dobaj, M. Seidl, and G. Macher. Architectural Patterns for Integrating AI Technology into Safety-Critical Systems. *European Conference on Pattern Languages of Programs (Euro-PLoP'21)*, 2021. doi: 10.1145/3489449.3490014.

[72] J. S. Edu, J. M. Such, and G. Suarez-Tangil. Smart Home Personal Assistants: A Security and Privacy Review. *ACM Computing Surveys*, 53(6), 2021. ISSN 15577341. doi: 10.1145/3412383.

[73] S. Eggimann, L. Mutzner, O. Wani, M. Y. Schneider, D. Spuhler, M. Moy De Vitry, P. Beutler, and M. Maurer. The Potential of Knowing More: A Review of Data-Driven Urban Water Management. *Environmental Science and Technology*, 51(5):2538–2553, 2017. ISSN 15205851. doi: 10.1021/acs.est.6b04267.

[74] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia. An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges. *IEEE Internet of Things Journal*, 5 (5):3758–3773, 2018. ISSN 23274662. doi: 10.1109/JIOT.2018.2844296.

[75] M. Elish and T. Hwang. *An AI Pattern Language*. 2016. ISBN 9781539033820. URL http://autonomy.datasociety.net/patternlanguage/.

[76] D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian. Runaway Feedback Loops in Predictive Policing. pages 1–12, 2017. URL http://arxiv.org/abs/1706.09847.

[77] M. Estevez and L. Ferrer. Study on the Fairness of Speaker Verification Systems on Underrepresented Accents in English. 2022. URL http://arxiv.org/abs/2204.12649.

[78] European Commission. Regulatory framework proposal on artificial intelligence, 2022. URL https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai.

[79] European Commission. European AI Alliance, 2022. URL https://futurium.ec.europa.eu/en/european-ai-alliance.

[80] European Commission. High-level expert group on artificial intelligence, 2022. URL https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai.

[81] R. Fang, S. Pouyanfar, Y. Yang, S. C. Chen, and S. S. Iyengar. Computational health informatics in the big data age: A survey. *ACM Computing Surveys*, 49(1), 2016. ISSN 15577341. doi: 10.1145/2932707.

[82] E. Farella, M. Rusci, B. Milosevic, and A. L. Murphy. Technologies for a thing-centric internet of things. *Proceedings - 2017 IEEE 5th International Conference on Future Internet of Things and Cloud, FiCloud 2017*, 2017-Janua:77–84, 2017. doi: 10.1109/FiCloud.2017.58.

[83] S. Fazelpour and M. De-Arteaga. Diversity in sociotechnical machine learning systems. *Big Data and Society*, 9(1), 1 2022. ISSN 20539517. doi: 10.1177/20539517221082027.

[84] Federica Laricchia. "Number of digital voice assistants in use worldwide from 2019 to 2024". URL https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/.

[85] G. Fenu, H. Lafhouli, and M. Marras. *Exploring Algorithmic Fairness in Deep Speaker Verification*, volume 12252 LNCS. Springer International Publishing, 2020. ISBN 9783030588106. doi: 10.1007/978-3-030-58811-3{_}6. URL http://dx.doi.org/10.1007/978-3-030-58811-3_6.

[86] R. Ferenc, . Beszédes, L. Fülöp, and J. Lele. Design pattern mining enhanced by machine learning. In *IEEE International Conference on Software Maintenance, ICSM*, volume 2005, pages 295–304, 2005. ISBN 0769523684. doi: 10.1109/ICSM.2005.40.

[87] L. Ferrer, M. McLaren, and N. Brümmer. A speaker verification backend with robust performance across conditions. *Computer Speech and Language*, 71(June 2021):1–23, 2022. ISSN 10958363. doi: 10.1016/j.csl.2021.101258.

[88] J. Fjeld, N. Achten, H. Hilligoss, A. Nagy, and M. Srikumar. Principled Artificial Intelligence: Mapping Consensus in Ethical and Rights-based Approaches to Principles for AI. Technical report, Berkman Klein Center for Internet & Society, 2020.

[89] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari. Enabling effective programming and flexible management of efficient body sensor network applications. *IEEE Transactions on Human-Machine Systems*, 43(1):115–133, 2013. ISSN 21682291. doi: 10.1109/TSMCC.2012.2215852.

[90] B. Friedman and H. Nissenbaum. Bias in computer systems. *Computer Ethics*, 14(3):215–232, 1996. doi: 10.4324/9781315259697-23.

[91] S. Furui. An Overview of Speaker Recognition Technology. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 1 – 9, 1994.

[92] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 2014. ISSN 15577341. doi: 10.1145/2523813.

[93] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman Publishing Co., Inc., USA, 1995. ISBN 0201633612.

[94] S. W. Gates, V. G. Perry, and P. M. Zorn. Automated underwriting in mortgage lending: Good news for the underserved? *Housing Policy Debate*, 13(2):369–391, 2002. ISSN 10511482. doi: 10.1080/10511482.2002.9521447.

[95] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. ISSN 15577317. doi: 10.1145/3458723.

[96] M. Ghorbanian, S. H. Dolatabadi, and P. Siano. Big Data Issues in Smart Grids: A Survey. *IEEE Systems Journal*, PP:1–12, 2019. ISSN 1932-8184. doi: 10.1109/jsyst.2019.2931879.

[97] A. Ghosh, L. Genuit, M. Reagan, D. Lamba, and W. H. Hsu. Characterizing Intersectional Group Fairness with Worst-Case Comparisons. Technical report, 2021. URL https://www.justice.gov/crt/equal-credit-opportunity-act-3.

[98] S. S. Gill, M. Xu, C. Ottaviani, P. Patros, R. Bahsoon, A. Shaghaghi, M. Golec, V. Stankovski, H. Wu, A. Abraham, M. Singh, H. Mehta, S. K. Ghosh, T. Baker, A. K. Parlikad, H. Lutfiyya, S. S. Kanhere, R. Sakellariou, S. Dustdar, O. Rana, I. Brandic, and S. Uhlig. AI for next generation computing: Emerging trends and future directions. *Internet of Things*, 19(March):100514, 2022. ISSN 25426605. doi: 10.1016/j.iot.2022.100514. URL https://doi.org/10.1016/j.iot.2022.100514.

[99] S. Gordillo, F. Balaguer, C. Mostaccio, and F. Das Neves. Developing GIS applications with objects: A design patterns approach. *GeoInformatica*, 3(1):7–32, 1999. ISSN 13846175. doi: 10.1023/A:1009809511770.

[100] B. Green. Escaping the Impossibility of Fairness: From Formal to Substantive Algorithmic Fairness. *Philosophy and Technology*, 35(4), 12 2022. ISSN 22105441. doi: 10.1007/s13347-022-00584-6.

[101] C. S. Greenberg, L. P. Mason, S. O. Sadjadi, and D. A. Reynolds. Two decades of speaker recognition evaluation at the national institute of standards and technology. *Computer Speech and Language*, 60, 2020. ISSN 10958363. doi: 10.1016/j.csl.2019.101032.

[102] C. Greer, M. Burns, D. Wollman, and E. Griffor. Cyber-Physical Systems and Internet of Things NIST Special Publication 1900-202. Technical report, National Institue of Standards and Technology, 2019.

[103] S. Gregor and A. R. Hevner. Positioning and presenting design science research for maximum impact. *MIS Quarterly: Management Information Systems*, 37(2):337–355, 2013. ISSN 21629730. doi: 10.2 5300/MISQ/2013/37.2.01.

[104] N. Grgi´c-Hlača, M. B. Zafar, K. P. Gummadi, and A. Weller. The Case for Process Fairness in Learning: Feature Selection for Fair Decision Making. In *Symposium on Machine Learning and the Law at the 29th Conference on Neural Information Processing Systems*, 2016.

[105] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 5 2018. ISSN 00313203. doi: 10.1016/j.patcog.2017.10.013.

[106] A. Gómez Ortega, J. Bourgeois, W. Hutiri, and G. Kortuem. Beyond Data Transactions: A Framework for Meaningfully Informed Data Donation. *AI & Society*, 2023. doi: 10.1007/s00146-023-01755-5.

[107] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pages 1–14, 2016.

[108] J. H. Hansen and T. Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015. ISSN 10535888. doi: 10.1109/MSP.2015.2462851.

[109] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, (Nips):3323–3331, 2016. ISSN 10495258.

[110] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009. ISBN 978-0-387-84857-0. doi: 10.1198/jasa.2004.s339.

[111] D. Hay. *Data Model Patterns: Conventions of Thought*. Dorset House Pub., 1996. ISBN 9780932633293.

[112] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw. Streaming Small-Footprint Keyword Spotting Using Sequence-to-Sequence Models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017. ISBN 9781509047888.

[113] K. Hechmi, T. N. Trong, V. Hautamaki, and T. Kinnunen. VoxCeleb Enrichment for Age and Gender Recognition. In *Automatic Speech Recognition and Understanding Workshop (ASRU)*, Cartagena, Colombia, 2021. IEEE. doi: 10.1109/ASRU51503.2021.9688085.

[114] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer. End-to-End Text-Dependent Speaker Verification. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE, 2016. ISBN 9781479999880.

[115] J. Henderson, S. Sharma, A. Gee, V. Alexiev, S. Draper, C. Marin, Y. Hinojosa, C. Draper, M. Perng, L. Aguirre, M. Li, S. Rouhani, S. Consul, S. Michalski, A. Prasad, M. Chutani, A. Kumar, S. Alam, P. Kandarpa, B. Jesudasan, C. Lee, M. Criscolo, S. Williamson, M. Sanchez, and J. Ghosh. Certifai: A Toolkit for Building Trust in AI Systems. In *International Joint Conference on Artificial Intelligence (IJCAI) Demonstrations Track - Proceedings*, 2020.

[116] H. S. Heo, B. J. Lee, J. Huh, and J. S. Chung. Clova baseline system for the VoxCeleb speaker recognition challenge 2020. 2020. URL https://arxiv.org/abs/2009.14153.

[117] A. Hevner and S. Chatterjee. *Design Research in Information Systems*. Springer, 2010. ISBN 978-1-4419-5652-1. doi: 10.1007/978-1-4419-5653-8. URL http://www.springer.com/series/6157.

[118] A. R. Hevner, S. T. March, J. Park, S. Ram, and S. Ram. Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105, 2004. doi: 10.2307/25148625. URL https://www.jstor.org/stable/25148625.

[119] High Level Expert Group on Artificial Intelligence. Ethics Guidelines for Trustworthy AI. Technical report, European Commission, 2019.

[120] T. Higuchi, M. Ghasemzadeh, K. You, and C. Dhir. Stacked 1D convolutional networks for end-to-end small footprint voice trigger detection. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020. doi: 10.21437/Interspeech.2020-2763.

[121] A. L. Hoffmann. Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse. *Information Communication and Society*, 22(7):900–915, 6 2019. ISSN 14684462. doi: 10.1080/1369118X.2019.1573912.

[122] K. Holstein, J. W. Vaughan, H. Daumé, M. Dudík, and H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–16, 2019. doi: 10.1145/3290605.3300830.

[123] S. Hooker, N. Moorosi, G. Clark, S. Bengio, and E. Denton. Characterising Bias in Compressed Models, 2020. URL https://arxiv.org/abs/2010.03058.

[124] J. Horkoff. Non-functional requirements for machine learning: Challenges and new directions. *Proceedings of the IEEE International Conference on Requirements Engineering*, 2019-Septe:386–391, 2019. ISSN 23326441. doi: 10.1109/RE.2019.00050.

[125] M. Hort and F. Sarro. *Privileged and Unprivileged Groups : An Empirical Study on the Impact of the Age Attribute on Fairness*, volume 1. Association for Computing Machinery, 2022. ISBN 9781450392921. doi: 10.1145/3524491.3527308.

[126] G. Huang, D. Chen, T. Li, F. Wu, L. Van Der Maaten, and K. Weinberger. Multi-scale dense networks for resource efficient image classification. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–14, 2018.

[127] B. Hutchinson, A. Smart, A. Hanna, E. Denton, C. Greer, O. Kjartansson, P. Barnes, and M. Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 560–575, 2021. doi: 10.1145/3442188.3445918.

[128] W. Hutiri and A. Y. Ding. Bias in Automated Speaker Recognition. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, pages 230–247, Seoul, Republic of Korea, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533089.

[129] W. Hutiri and A. Yi Ding. Towards Trustworthy Edge Intelligence: Insights from Voice-Activated Services. In *2022 IEEE International Conference on Services Computing (SCC)*, pages 239–248. IEEE Computer Society, 2022. ISBN 978-1-6654-8146-5. doi: 10.1109/SCC55611.2022.00043.

[130] W. Hutiri, L. Gorce, and A. Y. Ding. Design Guidelines for Inclusive Speaker Verification Evaluation Datasets. In *Interspeech 2022*, Incheon, Republic of Korea, 2022. International Speech Communication Association. doi: 10.21437/Interspeech.2022-10799.

[131] W. Hutiri, A. Y. Ding, F. Kawsar, and A. Mathur. Tiny, Always-on and Fragile: Bias Propagation through Design Choices in On-device Machine Learning Workflows. *ACM Transactions on Software Engineering and Methodology*, 4 2023. ISSN 1049-331X. doi: 10.1145/3591867.

[132] J. E. Ibarra-Esquer, F. F. González-Navarro, B. L. Flores-Rios, L. Burtseva, and M. A. Astorga-Vargas. Tracking the evolution of the internet of things concept across different application domains. *Sensors (Switzerland)*, 17(6):1–24, 2017. ISSN 14248220. doi: 10.3390/s17061379.

[133] A. Z. Jacobs and H. Wallach. Measurement and fairness. *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 375–385, 2021. ISSN 2331-8422. doi: 10.1145/3442188.3445901.

[134] M. Janssen, P. Brous, E. Estevez, L. S. Barbosa, and T. Janowski. Data governance: Organizing data for trustworthy Artificial Intelligence. *Government Information Quarterly*, 37(3), 7 2020. ISSN 0740624X. doi: 10.1016/j.giq.2020.101493.

[135] M. Jin, C. J. T. Ju, Z. Chen, Y.-C. Liu, J. Droppo, and A. Stolcke. Adversarial Reweighting for Speaker Verification Fairness. In *Proc. Interspeech*, Incheon, Korea, 2022. doi: 10.21437/Interspeech.2022-10948.

[136] B. Johnson and Y. Brun. Fairkit-learn. In *International Conference on Software Engineering Companion (ICSE-Companion) - Proceedings*, pages 70–74. Association for Computing Machinery (ACM), 5 2022. doi: 10.1145/3510454.3516830.

[137] J. Joo and K. Kärkkäinen. Gender Slopes: Counterfactual Fairness for Computer Vision Models by Attribute Manipulation. In *FATE/MM 2020 - Proceedings of the 2nd International Workshop on Fairness, Accountability, Transparency and Ethics in Multimedia*, pages 1–5. Association for Computing Machinery, Inc, 10 2020. ISBN 9781450381482. doi: 10.1145/3422841.3423533.

[138] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science Magazine*, 349(6245):255–260, 2015. doi: 10.1126/science.aac4520.

[139] Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner. Machine Bias, 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[140] D. Kahneman. Judgement and Decision Making: A Personal View. *Psychological Science*, 2(3), 1991.

[141] D. Kahneman, J. L. Knetsch, and R. H. Thaler. Fairness and the Assumptions of Economics. *The Journal of Business*, 59(4):285–300, 1986. URL http://www.jstor.org/stable/2352761.

[142] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and Open Problems in Federated Learning. pages 1–105, 2019. URL http://arxiv.org/abs/1912.04977.

[143] D. Katare, N. Kourtellis, S. Park, D. Perino, M. Janssen, and A. Y. Ding. Bias Detection and Generalization in AI Algorithms on Edge for Autonomous Driving. In *Proceedings - 2022 IEEE/ACM 7th Symposium on Edge Computing, SEC 2022*, pages 342–348. Institute of Electrical and Electronics Engineers Inc., 2022. ISBN 9781665486118. doi: 10.1109/SEC54971.2022.00050.

[144] F. Kawsar, C. Min, A. Mathur, A. Montanari, O. Amft, and K. Van Laerhoven. Earables for personal-scale behavior analytics. *IEEE Pervasive Computing*, 17(3):83–89, 2018. ISSN 15582590. doi: 10.1109/MPRV.2018.03367740.

[145] Z. Khan and Y. Fu. One label, one billion faces: Usage and consistency of racial categories in computer vision. In *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 587–597. Association for Computing Machinery, Inc, 3 2021. ISBN 9781450383097. doi: 10.1145/3442188.3445920.

[146] E. Khoury, B. Vesnicer, J. Franco-Pedroso, R. Violato, Z. Boulkcnafet, L. M. Mazaira Fernandez, M. Diez, J. Kosmala, H. Khemiri, T. Cipr, R. Saeidi, M. Gunther, J. Zganec-Gros, R. Z. Candil, F. Simoes, M. Bengherabi, A. Alvarez Marquina, M. Penagarikano, A. Abad, M. Boulayemen, P. Schwarz, D. Van Leeuwen, J. Gonzalez-Dominguez, M. U. Neto, E. Boutellaa, P. G. Vilda, A. Varona, D. Petrovska-Delacretaz, P. Matejka, J. Gonzalez-Rodriguez, T. Pereira, F. Harizi, L. J. Rodriguez-Fuentes, L. E. Shafey, M. Angeloni, G. Bordel, G. Chollet, and S. Marcel. The 2013 speaker recognition evaluation in mobile environment. *Proceedings - 2013 International Conference on Biometrics, ICB 2013*, 2013. doi: 10.1109/ICB.2013.6613025.

[147] A. Khritankov. Hidden Feedback Loops in Machine Learning Systems: A Simulation Model and Preliminary Results. In *Lecture Notes in Business Information Processing*, volume 404, pages 54–65. Springer Science and Business Media Deutschland GmbH, 2021. ISBN 9783030658533. doi: 10.1007/978-3-030-65854-0{_}5.

[148] R. Kienzler and I. Nesic. CLAIMED, a visual and scalable component library for Trusted AI. Technical report, 2020.

[149] C. Kim and R. M. Stern. Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(7):1315–1329, 2016. ISSN 23299290. doi: 10.1109/TASLP.2016.2545928.

[150] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. ISSN 15324435.

[151] T. Kinnunen and H. Li. An Overview of Text-Independent Speaker Recognition : from Features to Supervectors. *Speech Communication*, 52(1):12, 2009. doi: 10.1016/j.specom.2009.08.009.

[152] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, S. Mullainathan, D. Abrams, M. Alsdorf, M. Cohen, A. Crohn, G. R. Cusick, T. Dierks, J. Donohue, M. Dupont, M. Egan, E. Glazer, J. Gottschall, N. Hess, K. Kane, L. Kellam, A. Lascala-Gruenewald, C. Loeffler, A. Milgram, L. Raphael, C. Rohlfs, D. Rosenbaum, T. Salo, A. Shleifer, A. Sojourner, J. Sowerby, C. Sunstein, M. Sviridoff, E. Turner, and J. Wasilewski. Human Decisions and Machine Predictions. 2017. URL http://www.nber.org/papers/w23180.

[153] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Leibniz International Proceedings in Informatics, LIPIcs*, volume 67. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 11 2017. ISBN 9783959770293. doi: 10.4230/LIPIcs.ITCS.2017.43.

[154] J. Kleinberg, J. Ludwig, S. Mullainathan, and C. R. Sunstein. Discrimination in the Age of Algorithms. *Journal of Legal Analysis*, 10:113–174, 12 2018. doi: 10.1093/jla/laz001.

[155] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel. Racial disparities in automated speech recognition. *PNAS*, 117(14):7684–7689, 2020. doi: 10.1073/pnas.1915768117/-/DCSupplemental.y.

[156] J. Koppenjan and J. Groenewegen. Institutional design for complex technological systems. *International Journal of Technology, Policy and Management*, 5(3):240–257, 2005. ISSN 17415292. doi: 10.1504/IJTPM.2005.008406.

[157] K. S. Krishnapriya, K. Vangara, M. C. King, V. Albiero, and K. Bowyer. Characterizing the variability in face recognition accuracy relative to race. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019-June:2278–2285, 2019. ISSN 21607516. doi: 10.1109/CVPRW.2019.00281.

[158] A. Kumar, T. Braud, S. Tarkoma, and P. Hui. Trustworthy AI in the Age of Pervasive Computing and Big Data. *2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2020*, 2020. doi: 10.1109/PerComWorkshops48775.2020.9156127.

[159] M. Kusner, J. Loftus, C. Russel, and R. Silva. Counterfactual Fairness. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017. URL http://papers.nips.cc/paper/6995-counterfactual-fairness.

[160] A. Lakhdari and A. Bouguettaya. *Fairness-Aware Crowdsourcing of IoT Energy Services*, volume 13121 LNCS. Springer International Publishing, 2021. ISBN 9783030914301. doi: 10.1007/978-3-030-91431-8{_}22.

[161] V. Lakshmanan, S. Robinson, and M. Munn. *Machine Learning Design Patterns*. O'Reilly Media, 2020. ISBN 9781098115784.

[162] N. K. Lankton, D. Harrison Mcknight, and J. Tripp. Technology, humanness, and trust: Rethinking trust in technology. *Journal of the Association for Information Systems*, 16(10):880–918, 2015. ISSN 15583457. doi: 10.17705/1jais.00411.

[163] K. Lasse Lueth. State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating, 2018. URL https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/.

[164] J. Lau, B. Zimmerman, and F. Schaub. Alexa, Are You Listening? *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–31, 2018. ISSN 2573-0142. doi: 10.1145/3274371.

[165] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. ISSN 14764687. doi: 10.1038/nature14539.

[166] M. S. A. Lee and J. Singh. The landscape and gaps in open source fairness toolkits. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, 5 2021. ISBN 9781450380966. doi: 10.1145/3411764.3445261.

[167] M. Levi and L. Stoker. Political Trust and Trustworthiness. *Annual Review of Political Science*, 3(1992):475–507, 2000.

[168] S. Li, L. D. Xu, and S. Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015. ISSN 13873326. doi: 10.1007/s10796-014-9492-7.

[169] L. Liang, H. Ye, and G. Y. Li. Toward Intelligent Vehicular Networks: A Machine Learning Framework. *IEEE Internet of Things Journal*, 6(1):124–135, 2019. ISSN 23274662. doi: 10.1109/JIOT.2018.2872122.

[170] Y. Liang, D. O'Keeffe, and N. Sastry. PAIGE: Towards a hybrid-edge design for privacy-preserving intelligent personal assistants. *EdgeSys 2020 - Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2020*, pages 55–60, 2020. doi: 10.1145/3378679.3394536.

[171] L. Liao, H. Li, W. Shang, and L. Ma. An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks. *ACM Transactions on Software Engineering and Methodology*, 31(3):1–40, 2022. ISSN 1049-331X. doi: 10.1145/3506695.

[172] L. Liebenwein, C. Baykal, B. Carter, D. Gifford, and D. Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *Proceedings of Machine Learning and Systems*, 3:93–138, 2021.

[173] H. Liu, J. Dacon, W. Fan, H. Liu, Z. Liu, and J. Tang. Does Gender Matter? Towards Fairness in Dialogue Systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4403–4416. Online, 2020.

[174] J. Liu, S. Tripathi, U. Kurup, and M. Shah. Pruning Algorithms to Accelerate Convolutional Neural Networks for Edge Applications: A Survey. 2020. URL http://arxiv.org/abs/2005.04275.

[175] S. Liu and L. N. Vicente. Accuracy and fairness trade-offs in machine learning: a stochastic multi-objective approach. *Computational Management Science*, 19(3):513–537, 2022. ISSN 16196988. doi: 10.1007/s10287-022-00425-z.

[176] X. Liu, M. Sahidullah, and T. Kinnunen. A comparative Re-assessment of feature extractors for deep speaker embeddings. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020-Octob:3221–3225, 2020. ISSN 19909772. doi: 10.21437/Interspeech.2020-1765.

[177] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Douglas, and C. Sanderson. Software engineering for Responsible AI: An empirical study and operationalised patterns. pages 241–242, 2022. doi: 10.1109/icse-seip55303.2022.9793864.

[178] L. Lucy and D. Bamman. Gender and Representation Bias in GPT-3 Generated Stories. In *Proceedings of the 3rd Workshop on Narrative Understanding*, pages 48–55. ACL, 2021.

[179] K. Lum, Y. Zhang, and A. Bower. De-biasing "bias" measurement. In *FAccT '22: Conference on Fairness, Accountability, and Transparency*, pages 379–389, Seoul, Republic of Korea, 2022. ACM. doi: 10.1145/3531146.3533105.

[180] K. Makhlouf, S. Zhioua, and C. Palamidessi. On the Applicability of Machine Learning Fairness Notions. *ACM SIGKDD Explorations Newsletter*, 23(1):14–23, 2021. ISSN 1931-0145. doi: 10.1145/3468507.3468511.

[181] P. Mallozzi, P. Pelliccione, A. Knauss, C. Berger, and N. Mohammadiha. Autonomous Vehicles: State of the Art, Future Trends, and Challenges. *Automotive Systems and Software Engineering*, pages 347–367, 2019. doi: 10.1007/978-3-030-12157-0{_}16.

[182] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke. Feedback Loop and Bias Amplification in Recommender Systems. In *International Conference on Information and Knowledge Management, Proceedings*, pages 2145–2148. Association for Computing Machinery, 10 2020. ISBN 9781450368599. doi: 10.1145/3340531.3412152.

[183] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET Curve in Assessment of Detection Task Performance. Technical report, National Institute of Standards and Technology (NIST), Gaithersburg MD, 1997.

[184] Martin Fowler. *Analysis Patterns: Reusable Object Models*. Addison-Wesley, 1997. ISBN 0201895420.

[185] A. Mathur, N. D. Lane, S. Bhattacharya, A. Boran, C. Forlivesi, and F. Kawsar. DeepEye: Resource efficient local execution of multiple deep vision models using wearable commodity hardware. *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys 2017*, pages 68–81, 2017. doi: 10.1145/3081333.3081359.

[186] A. Mathur, T. Zhang, S. Bhattacharya, P. Velickovic, L. Joffe, N. D. Lane, F. Kawsar, and P. Lió. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 200–211. IEEE, 2018.

[187] R. C. Mayer, J. H. Davis, and F. David Schoorman. An Integrative Model of Organizational Trust. *The Academy of Management Review*, 20(3):709–734, 1995. doi: 10.2307/258792. URL https://www.jstor.org/stable/258792.

[188] M. Mazumder, S. Chitlangia, C. Banbury, Y. Kang, J. Ciro, K. Achorn, D. Galvez, M. Sabini, P. Mattson, D. Kanter, G. Diamos, P. Warden, J. Meyer, and V. J. Reddi. Multilingual Spoken Words Corpus. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 preproceedings (NeurIPS Datasets and Benchmarks 2021)*, number 1, 2021.

[189] K. Mcauliffe, P. R. Blake, N. Steinbeis, and F. Warneken. The Developmental Foundations of Human Fairness. *Nature Human Behaviour*, 1(0042), 2017. doi: 10.1038/s41562-016-0042.

[190] D. H. McKnight, V. Choudhury, and C. Kacmar. Developing and Validating Trust Measures for e-Commerce: An Integrative Typology. *Information Systems Research*, 13(3):334–359, 2002. ISSN 1526-5536.

[191] M. McLaren, L. Ferrer, D. Castan, and A. Lawson. The Speakers in the Wild (SITW) speaker recognition database. In *Proc. Interspeech*, San Francisco, CA, USA, 2016. pdfs.semanticscholar.org. doi: 10.21437/Interspeech.2016-1129.

[192] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6), 8 2019. doi: 10.1145/3457607.

[193] M. Merenda, C. Porcaro, and D. Iero. Edge Machine Learning for AI-enabled IoT devices: A Review. *Sensors*, 20(9):1–34, 2020. ISSN 14248220. doi: 10.3390/s20092533.

[194] Merriam-Webster.com Dictionary. "discrimination", . URL https://www.merriam-webster.com/dictionary/discrimination.

[195] Merriam-Webster.com Dictionary. "pattern", . URL https://www.merriam-webster.com/dictionary/pattern.

[196] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, (Figure 2):220–229, 2019. doi: 10.1145/3287560.3287596.

[197] S. Mitchell, E. Potash, S. Barocas, A. D'Amour, and K. Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8:141–163, 2021. ISSN 2326831X. doi: 10.1146/annurev-statistics-042720-125902.

[198] T. M. Mitchell. The Discipline of Machine Learning. 2006. URL http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf.

[199] B. Mittelstadt, S. Wachter, and C. Russell. The Unfairness of Fair Machine Learning: Levelling down and strict egalitariansim by default. *Michigan Technology Law Review*, 2023.

[200] B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi. The ethics of algorithms: Mapping the debate. *Big Data and Society*, 3(2):1–21, 2016. ISSN 20539517. doi: 10.1177/2053951716679679.

[201] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys and Tutorials*, 20(4):2923–2960, 2018. ISSN 1553877X. doi: 10.1109/COMST.2018.2844341.

[202] M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gómez-Romero, and M. J. Martin-Bautista. Data science for building energy management: A review. *Renewable and Sustainable Energy Reviews*, 70(December 2016):598–609, 2017. doi: 10.1016/j.rser.2016.11.132.

[203] A. Montanari, M. Sharma, D. Jenkus, M. Alloulah, L. Qendro, and F. Kawsar. eperceptive: energy reactive embedded intelligence for batteryless sensors. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 382–394, 2020.

[204] M. Morrás. BBVA Mexico allows its pensioner customers to provide proof of life from home thanks to Veridas voice biometrics, 2021. URL https://veridas.com/en/bbva-mexico-allows-pensioner-customers-provide-proof-of-life-from-home/.

[205] A. Musil, J. Musil, D. Weyns, T. Bures, H. Muccini, and M. Sharaf. Patterns for Self-Adaptation in Cyber-Physical Systems. In S. Biffl, D. Gerhard, and A. Lüder, editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, chapter 13, pages 331–367. Springer International Publishing, 2017. ISBN 9783319563459. doi: 10.1007/978-3-319-56345-9.

[206] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad. Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE Transactions on Power Delivery*, 25 (2):1162–1171, 2010. ISSN 08858977. doi: 10.1109/TPWRD.2009.2030890.

[207] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Proc. Interspeech*, pages 2616–2620, Stockholm, Sweden, 2017. ISCA. doi: 10.21437/Interspeech.2017-950.

[208] A. Nagrani, J. S. Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman. VoxSRC 2020: The Second VoxCeleb Speaker Recognition Challenge. 2020. URL http://arxiv.org/abs/2012.06867.

[209] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech and Language*, 60:101027, 2020. ISSN 10958363. doi: 10.1016/j.csl.2019.101027. URL https://doi.org/10.1016/j.csl.2019.101027.

[210] A. Narayanan. 21 definitions of fairness and their politics., 2018. URL https://www.youtube.com/embed/jIXIuYdnyyk.

[211] A. Narayanan. Translation Tutorial: 21 definitions of fairness and their politics., 2018. URL https://facctconference.org/static/tutorials/narayanan-21defs18.pdf.

[212] D. J. Navarro, D. R. Foxcroft, and T. J. Faulkenberry. Factorial Anova. In *Learning Statistics with JASP: A Tutorial for Psychology Students and Other Beginners*, chapter Chapter 13, pages 327 – 380. online, 2019. doi: 10.1002/9781119121077.ch10. URL https://tomfaulkenberry.github.io/JASPbook/chapters/chapter13.pdf.

[213] Nicholas Confessore. "Cambridge Analytica and Facebook: The Scandal and the Fallout So Far". URL https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html.

[214] M. Nieke, L. Almstedt, and R. Kapitza. Edgedancer: Secure Mobile WebAssembly Services on the Edge. *EdgeSys 2021 - Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2021*, pages 13–18, 2021. doi: 10.1145/3434770.3459731.

[215] R. Nielsen. *Is European Union equality law capable of addressing multiple and intersectional discrimination yet?: Precautions against neglecting intersectional cases*, pages 31–51. Routledge, United Kingdom, 2008. ISBN 9780415457224. løbenummer: 086139.

[216] NIST. NIST 2019 Speaker Recognition Evaluation Plan. (1):1–7, 2019.

[217] NIST. NIST 2020 CTS Speaker Recognition Challenge Evaluation Plan. Technical report, 2020.

[218] L. Oakden-Rayner, G. Carneiro, J. Dunnmon, and C. Ré. Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging. In *Machine Learning for Health at NeurIPS*, 2019.

[219] L. O'Sullivan. How the law got it wrong with Apple Card, 2021. URL https://techcrunch.com/2021/08/14/how-the-law-got-it-wrong-with-apple-card/.

[220] K. Padh, D. Antognini, E. Lejal-Glaude, B. Faltings, and C. Musat. Addressing Fairness in Classification with a Model-Agnostic Multi-Objective Algorithm. *37th Conference on Uncertainty in Artificial Intelligence, UAI 2021*, (Uai):600–609, 2021.

[221] S. J. Park, C. Sigouin, J. Kreiman, P. Keating, J. Guo, G. Yeung, F.-Y. Kuo, and A. Alwan. Speaker Identity and Voice Quality: Modeling Human Responses and Automatic Speaker Recognition. In *Proc. Interspeech*, San Francisco, CA, USA, 2016. ISCA. doi: 10.21437/Interspeech.2016-523.

[222] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In *British Machine Vision Conference*, 2015.

[223] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[224] R. K. Pathinarupothi, P. Durga, and E. S. Rangan. IoT-based smart edge for global health: Remote monitoring with severity detection and alerts transmission. *IEEE Internet of Things Journal*, 6(2): 2449–2462, 2019. doi: 10.1109/JIOT.2018.2870068.

[225] Paul Mozur. "A Genocide Incited on Facebook, With Posts From Myanmar's Military". URL https://www.nytimes.com/2018/10/15/technology/myanmar-facebook-genocide.html.

[226] A. Paullada, I. D. Raji, E. M. Bender, E. Denton, and A. Hanna. Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11), 11 2021. doi: 10.1016/j.patter.2021.100336.

[227] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007. doi: 10.2753/MIS0742-1222240302.

[228] E. Peltonen, I. Ahmad, A. Aral, M. Capobianco, A. Y. Ding, F. Gil-Castiñeira, E. Gilman, E. Harjula, M. Jurmu, T. Karvonen, M. Kelanti, T. Leppänen, L. Lovén, T. Mikkonen, N. Mohan, P. Nurmi, S. Pirttikangas, P. Sroka, S. Tarkoma, and T. Yang. The Many Faces of Edge Intelligence. *IEEE Access*, 10:104769–104782, 2022. doi: 10.1109/ACCESS.2022.3210584.

[229] R. Peri, K. Somandepalli, and S. Narayanan. A study of bias mitigation strategies for speaker recognition. *Computer Speech and Language*, 79:1–31, 2023. doi: 10.1016/j.csl.2022.101481.

[230] D. Pessach and E. Shmueli. A Review on Fairness in Machine Learning. *ACM Computing Surveys*, 55 (3):1–44, 2022. doi: 10.1145/3494672.

[231] Peter Coad. Object-Oriented Patterns. *Communications of the ACM*, 35(9):152–159, 1992. doi: 10.1145/130994.131006.

[232] I. Philippow, D. Streitferdt, M. Riebisch, and S. Naumann. An approach for reverse engineering of design patterns. *Software and Systems Modeling*, 4(1):55–70, 2 2005. doi: 10.1007/s10270-004-0059-9.

[233] N. R. Prasad, S. Almanza-Garcia, and T. T. Lu. Anomaly detection. *Computers, Materials and Continua*, 14(1):1–22, 2009. ISSN 15462218. doi: 10.1145/1541880.1541882.

[234] ProKNX. Smart Home technology keeps people happy at home for longer, 2022. URL https://www.proknx.com/en/news/2020/smart-home-technology-keeps-people-happy-at-home-for-longer/.

[235] S. Qian, H. V. Pham, T. Lutellier, Z. Hu, J. Kim, L. Tan, Y. Yu, J. Chen, J. P. Morgan, and S. Shah. Are My Deep Learning Systems Fair? An Empirical Study of Fixed-Seed Training. In *35th Conference on Neural Information Processing Systems (NeurIPS)*, number NeurIPS, 2021.

[236] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li. The INTERSPEECH 2020 far-field speaker verification challenge. *Proc. Interspeech*, pages 3456–3460, 2020. doi: 10.21437/Interspeech.2020-1249.

[237] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 469–481, 2020. doi: 10.1145/3351095.3372828.

[238] I. D. Raji and J. Buolamwini. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial AI products. *AIES 2019 - Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 429–435, 2019. doi: 10.1145/3306618.3314244.

[239] I. D. Raji and G. Fried. About Face: A Survey of Facial Recognition Evaluation. In *AAAI 2020 Workshop on AI Evaluation*, 2021.

[240] I. D. Raji, T. Gebru, M. Mitchell, J. Buolamwini, J. Lee, and E. Denton. Saving Face: Investigating the ethical concerns of facial recognition auditing. *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 145–151, 2020. doi: 10.1145/3375627.3375820.

[241] R. Ramachandra, K. Raja, and C. Busch. Algorithmic Fairness in Face Morphing Attack Detection. *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACVW 2022*, pages 410–418, 2022. doi: 10.1109/WACVW54805.2022.00047.

[242] D. Ravi, C. Wong, B. Lo, and G.-z. Yang. A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices. *IEEE Journal of Biomedical and Health Infromatics*, 12(1):106–137, 2017. doi: 10.1109/JBHI.2016.2633287.

[243] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar. A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wireless Communications and Mobile Computing*, 2019, 2019. ISSN 15308677. doi: 10.1155/2019/3159762.

[244] S. Retalis, P. Georgiakakis, and Y. Dimitriadis. Eliciting design patterns for e-learning systems. *Computer Science Education*, 16(2):105–118, 2006. ISSN 17445175. doi: 10.1080/08993400600773323.

[245] B. Ricks and M. Surman. Creating Trustworthy AI a Mozilla whitepaper on challenges and opportunities in the AI era. Technical Report April, Mozilla Foundation, 4 2020.

[246] D. Riehle and H. Zullighoven. Understanding and using patterns in software development. *Theory and Practice of Object Systems*, 2(1):3–13, 1996. ISSN 10743227. doi: 10.1002/(SICI)1096-9942(1996)2:1<3::AID-TAPO1>3.0.CO;2-{\#}.

[247] A. Romanovsky and F. Ishikawa. *Trustworthy cyber-physical systems engineering*. CRC Press, 2016.

[248] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach 4th Edition*. Pearson Education Limited, USA, 2022.

[249] C. Rusti, A. Leschanowsky, C. Quinlan, M. Pnacek, L. Gorce, and W. Hutiri. Benchmark Dataset Dynamics, Bias and Privacy Challenges in Voice Biometrics Research. In *International Joint Conference on Biometrics (IJCB)*. IEEE, 2023.

[250] A. Saad Al-Sumaiti, M. H. Ahmed, and M. M. Salama. Smart home activities: A literature review. *Electric Power Components and Systems*, 42(3-4):294–305, 2014. ISSN 15325008. doi: 10.1080/15325008.2013.832439.

[251] T. N. Sainath and C. Parada. Convolutional neural networks for small-footprint keyword spotting. In *Proc. Interspeech*, volume 2015-Janua, pages 1478–1482, 2015. doi: 10.21437/interspeech.2015-352.

[252] S. Saito, Y. Ide, T. Nakano, and T. Ogawa. VocalTurk: Exploring feasibility of crowdsourced speaker identification. *Proc. Interspeech*, pages 2932–2936, 2021. doi: 10.21437/Interspeech.2021-464.

[253] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani. Aequitas: A Bias and Fairness Audit Toolkit. (2018), 2018. URL http://arxiv.org/abs/1811.05577.

[254] F. Samie, L. Bauer, and J. Henkel. IoT Technologies for Embedded Computing : A Survey. In *CODES/ISSS '16*, Pittsburgh, USA, 2016. ISBN 9781450344838.

[255] F. Samie, L. Bauer, and J. Henkel. From cloud down to things: An overview of machine learning in internet of things. *IEEE Internet of Things Journal*, 6(3):4921–4934, 2019. ISSN 23274662. doi: 10.1109/JIOT.2019.2893866.

[256] S. Schelter, F. Biessmann, T. Januschowski, D. Salinas, S. Seufert, and G. Szarvas. On Challenges in Machine Learning Model Management. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, pages 5–13, 2018.

[257] M. K. Scheuerman, J. M. Paul, and J. R. Brubaker. How computers see gender: An evaluation of gender classification in commercial facial analysis and image labeling services. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW), 2019. ISSN 25730142. doi: 10.1145/3359246.

[258] L. Schönherr, M. Golla, T. Eisenhofer, J. Wiele, D. Kolossa, and T. Holz. Unacceptable, where is my privacy? Exploring Accidental Triggers of Smart Speakers. 8 2020. URL http://arxiv.org/abs/2008.00508.

[259] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems*, 2015.

[260] K. Seaborn, N. P. Miyake, P. Pennefather, and M. Otake-Matsuura. Voice in human-agent interaction: A survey. *ACM Computing Surveys*, 54(4), 2021. ISSN 15577341. doi: 10.1145/3386867.

[261] K. Seaborn, N. P. Miyake, P. Pennefather, and M. Otake-Matsuura. Voice in human-agent interaction: A survey. *ACM Computing Surveys*, 54(4), 2021. ISSN 15577341. doi: 10.1145/3386867.

[262] K. Sharif and B. Tenbergen. Smart Home Voice Assistants: A Literature Survey of User Privacy and Security Vulnerabilities. *Complex Systems Informatics and Modeling Quarterly*, (24):15–30, 2020. doi: 10.7250/csimq.2020-24.02.

[263] H. Shen, Y. Yang, G. Sun, R. Langman, E. Han, J. Droppo, and A. Stolcke. Improving Fairness in Speaker Verification via Group-Adapted Fusion Network. pages 7077–7081, 2022. doi: 10.1109/icassp43922.2022.9747384.

[264] W. Sherchan, S. Nepal, and C. Paris. A survey of trust in social networks. *ACM Computing Surveys*, 45 (4), 8 2013. ISSN 03600300. doi: 10.1145/2501654.2501661.

[265] N. Shi and R. A. Olsson. Reverse engineering of design patterns from Java source code. In *Proceedings - 21st IEEE/ACM International Conference on Automated Software Engineering, ASE 2006*, pages 123–132, 2006. ISBN 0769525792. doi: 10.1109/ASE.2006.57.

[266] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 10 2016. ISSN 23274662. doi: 10.1109/JIOT.2016.2579198.

[267] J. Siebert, L. Joeckel, J. Heidrich, A. Trendowicz, K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, and M. Aoyama. Construction of a quality model for machine learning systems. *Software Quality Journal*, 2021. doi: 10.1007/s11219-021-09557-y.

[268] H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, third [2019] edition, 1996. ISBN 9780262537537.

[269] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–14, 2015.

[270] H. Singh, R. Singh, V. Mhasawade, and R. Chunara. Fairness violations and mitigation under covariate shift. *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 3–13, 2021. doi: 10.1145/3442188.3445865.

[271] R. Singh. *Profiling Humans from their Voice*. 2019. ISBN 9789811384028. doi: 10.1007/978-981-13-8403-5.

[272] D. Situnayake and J. Plunkett. *AI at the Edge*. O'Reilly Media, 2022. ISBN 9781098120207.

[273] S. Small, S. Khalid, P. Dhiman, S. Chan, D. Jackson, A. Doherty, and A. Price. Impact of reduced sampling rate on accelerometer-based physical activity monitoring and machine learning activity classification. *Journal for the Measurement of Physical Behaviour*, 4(4):298–310, 2021.

[274] T. Speicher, H. Heidari, N. Grgic-Hlaca, K. P. Gummadi, A. Singla, A. Weller, and M. B. Zafar. A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2239–2248, 2018. doi: 10.1145/3219819.3220046.

[275] V. D. Stanciu, M. V. Steen, C. Dobre, and A. Peter. Privacy-Preserving Crowd-Monitoring Using Bloom Filters and Homomorphic Encryption. *EdgeSys 2021 - Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2021*, pages 37–42, 2021. doi: 10.1145/3434770.3459735.

[276] S. Stumpf, L. Strappelli, S. Ahmed, Y. Nakao, A. Naseer, G. D. Gamba, and D. Regoli. Design Methods for Artificial Intelligence Fairness and Transparency. *CEUR Workshop Proceedings*, 2903, 2021. ISSN 16130073.

[277] H. Suresh and J. Guttag. A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. In *EAAMO '21: Equity and Access in Algorithms, Mechanisms, and Optimization*, 2021. ISBN 9781450385534.

[278] TAILOR. TAILOR – A Network of Research Excellence Centres, 2022. URL https://tailor-network.eu/.

[279] M. Take, S. Alpers, C. Becker, C. Schreiber, and A. Oberweis. AI systems development with design patterns. Technical report, 2021.

[280] J. Tang, D. Sun, S. Liu, and J.-L. Gaudiot. Enabling Deep Learning on IoT Devices. *IEEE Computer*, pages 92 – 96, 2017.

[281] R. Tatman. Gender and Dialect Bias in YouTube's Automatic Captions. pages 53–59, 2017. doi: 10.18653/v1/w17-1606.

[282] R. Tatman and C. Kasten. Effects of talker dialect, gender & race on accuracy of bing speech and youtube automatic captions. *Proc. Interspeech*, pages 934–938, 2017. doi: 10.21437/Interspeech.2017-1746.

[283] I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, A. Yuan, and G. Research. The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118. ACL, 2020. doi: 10.18653/v1/2020.emnlp-demos.15.

[284] The White House. Blueprint for an AI Bill of Rights, 2022. URL https://www.whitehouse.gov/ostp/ai-bill-of-rights/.

[285] E. Toreini, M. Aitken, K. Coopamootoo, K. Elliott, C. G. Zelaya, and A. van Moorsel. The relationship between trust in AI and trustworthy machine learning technologies. In *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 272–283. Association for Computing Machinery, Inc, 1 2020. ISBN 9781450369367. doi: 10.1145/3351095.3372834.

[286] W. Toussaint and A. Y. Ding. Machine Learning Systems in the IoT: Trustworthiness Trade-offs for Edge Intelligence. *2020 IEEE 2nd International Conference on Cognitive Machine Intelligence (CogMI 2020)*, pages 177–184, 2020. doi: 10.1109/CogMI50398.2020.00030.

[287] W. Toussaint and D. Moodley. Comparison of clustering techniques for residential load profiles in South Africa. In *Proceedings of the South African Forum for AI Research*, Cape Town, 2019. URL http://ceur-ws.org/Vol-2540/FAIR2019_paper_55.pdf.

[288] W. Toussaint and D. Moodley. Clustering Residential Electricity Consumption Data to Create Archetypes that Capture Household Behaviour in South Africa. *South African Computer Journal*, 32 (2):1–34, 2020. ISSN 23137835. doi: 10.18489/SACJ.V32I2.845.

[289] W. Toussaint and D. Moodley. Identifying optimal clustering structures for residential energy consumption patterns using competency questions. In *SAICSIT '20: Conference of the South African Institute of Computer Scientists and Information Technologists 2020*, 2020. ISBN 9781450388474. doi: 10.1145/3410886.3410887.

[290] W. Toussaint and D. Moodley. Using competency questions to select optimal clustering structures for residential energy consumption patterns. In *International Conference on Learning Representations (ICLR) 2020 Workshop on Machine Learning in Real Life*, 2020. URL http://arxiv.org/abs/2006.00934.

[291] W. Toussaint, D. Van Veen, C. Irwin, Y. Nachmany, M. Barreiro-Perez, E. Díaz-Peláez, S. G. de Sousa, L. Millán, P. L. Sánchez, A. Sánchez-Puente, J. Sampedro-Gómez, P. I. Dorado-Díaz, and V. Vicente-Palacios. Design Considerations for High Impact, Automated Echocardiogram Analysis. In *International Conference on Machine Learning (ICML) 2020 ML for Global Health Workshop*, 2020. URL http://arxiv.org/abs/2006.06292.

[292] W. Toussaint, A. Mathur, A. Y. Ding, and F. Kawsar. Characterising the Role of Pre-Processing Parameters in Audio-based Embedded Machine Learning. In *The 3rd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT '21)*, pages 439–445, Coimbra, Portugal, 2021. ACM. ISBN 9781450390972. doi: 10.1145/3485730.3493448.

[293] G. Trencher. Towards the smart city 2.0: Empirical evidence of using smartness as a tool for tackling social challenges. *Technological Forecasting and Social Change*, 142:117–128, 5 2019. ISSN 00401625. doi: 10.1016/j.techfore.2018.07.033.

[294] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei. LDP-Fed: Federated Learning with Local Differential Privacy. In *EdgeSys 2020 - Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2020*, pages 61–66, 2020. ISBN 9781728164328. doi: 10.1109/ISIT44484.2020.9174426.

[295] M. C. Tschantz. What is Proxy Discrimination? In *Proceedings of the Conference on Fairness, Accountability and Transparency (FAccT)*, pages 1993–2003, Seoul, Republic of Korea, 2022. ACM. doi: 10.1145/3531146.3533242.

[296] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni. Model compression applied to small-footprint keyword spotting. *Proc. Interspeech*, pages 1878–1882, 2016. doi: 10.21437/Interspeech.2016-1393.

[297] L. Tuggener, M. Amirian, F. Benites, P. von Däniken, P. Gupta, F.-P. Schilling, and T. Stadelmann. Design Patterns for Resource-Constrained Automated Deep-Learning Methods. *Ai*, 1(4):510–538, 2020. doi: 10.3390/ai1040031.

[298] J. P. Tuohy. Amazon Alexa's new elder care service launches today, 2021. URL https://www.theverge.com/2021/12/7/22822026/amazon-alexa-together-elder-care-price-features-release-date.

[299] C. J. Turillo, R. Folger, J. J. Lavelle, E. E. Umphress, J. O. Gee, and A. B. Freeman. Is virtue its own reward? Self-sacrificial decisions for the sake of fairness. *Organizational Behavior and Human Decision Processes*, 89:839–865, 2002.

[300] A. Tversky and D. Kahneman. Judgement under Uncertainty: Heuristics and Biases. *Science*, 185, 1974.

[301] I. van de Poel. Embedding Values in Artificial Intelligence (AI) Systems. *Minds and Machines*, 30(3): 385–409, 2020. doi: 10.1007/s11023-020-09537-4.

[302] W. Van Der Aalst, A. Ter Hofstede, B. Kiepuszewski, and A. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14:5–51, 2003. doi: 10.1007/978-3-030-26253-2{_}2.

[303] B. Varghese, E. De Lara, A. Y. Ding, C. H. Hong, F. Bonomi, S. Dustdar, P. Harvey, P. Hewkin, W. Shi, M. Thiele, and P. Willis. Revisiting the Arguments for Edge Computing Research. *IEEE Internet Computing*, 25(5):36–42, 2021. ISSN 19410131. doi: 10.1109/MIC.2021.3093924.

[304] S. Vasudevan and K. Kenthapadi. LiFT: A Scalable Framework for Measuring Fairness in ML Applications. In *International Conference on Information and Knowledge Management, Proceedings*, pages 2773–2780. Association for Computing Machinery, 10 2020. ISBN 9781450368599. doi: 10.1145/3340531.3412705.

[305] J. Venkatesh, B. Aksanli, C. S. Chan, A. S. Akyurek, and T. S. Rosing. Modular and Personalized Smart Health Application Design in a Smart City Environment. *IEEE Internet of Things Journal*, 5(2):614–623, 2018. ISSN 23274662. doi: 10.1109/JIOT.2017.2712558.

[306] S. Verma and J. Rubin. Fairness definitions explained. In *Proceedings - International Conference on Software Engineering*, pages 1–7, 2018. ISBN 9781450357463. doi: 10.1145/3194770.3194776.

[307] H. Villamizar, M. Kalinowski, and H. Lopes. A Catalogue of Concerns for Specifying Machine Learning-Enabled Systems. 2022. URL http://arxiv.org/abs/2204.07662.

[308] S. Wachter, B. Mittelstadt, and C. Russell. Bias Preservation in Machine Learning : The Legality of Fairness Metrics Under EU Non- Discrimination Law. *West Virginia Law Review*, pages 1–51, 2021.

[309] A. Wahyudi, G. Kuk, and M. Janssen. A Process Pattern Model for Tackling and Improving Big Data Quality. *Information Systems Frontiers*, 20(3):457–469, 6 2018. ISSN 15729419. doi: 10.1007/s10796-017-9822-7.

[310] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156, 2018. doi: 10.1016/j.jmsy.2018.01.003.

[311] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019. doi: 10.1016/j.patrec.2018.02.010.

[312] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika, and A. Gruenstein. VoiceFilter-Lite: Streaming targeted voice separation for on-device speech recognition. In *Interspeech 2020*, pages 2677–2681, 2020.

[313] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous. Trainable frontend for robust and far-field keyword spotting. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (1):5670–5674, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7953242.

[314] Y. Wang, Q. Chen, T. Hong, and C. Kang. Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid*, 10(3):3125–3148, 2019. ISSN 19493053. doi: 10.1109/TSG.2018.2818167.

[315] Z. Wang, H. Song, D. W. Watkins, K. G. Ong, P. Xue, Q. Yang, and X. Shi. Cyber-physical systems for water sustainability: Challenges and opportunities. *IEEE Communications Magazine*, 53(5):216–222, 2015. ISSN 01636804. doi: 10.1109/MCOM.2015.7105668.

[316] Z. Wang, K. Qinami, C. Karakozis, K. Genova, P. Nair, K. Hata, and O. Russakovsky. Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8919–8928, 2020.

[317] P. Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. 2018. URL https://arxiv.org/abs/1804.03209.

[318] H. Washizaki, F. Khomh, Y. G. Gueheneuc, H. Takeuchi, N. Natori, T. Doi, and S. Okuda. Software-Engineering Design Patterns for Machine Learning Applications. *Computer*, 55(3):30–39, 3 2022. ISSN 15580814. doi: 10.1109/MC.2021.3137227.

[319] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 1 2020. ISSN 19410506. doi: 10.1109/TVCG.2019.2934619.

[320] A. Whitmore, A. Agarwal, and L. Da Xu. The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, 2015. ISSN 15729419. doi: 10.1007/s10796-014-9489-2.

[321] M. Whittaker, K. Crawford, R. Dobbe, G. Fried, E. Kaziunas, V. Mathur, S. Myers West, R. Richardson, J. Schultz, and O. Schwartz. AI Now Report 2018. Technical report, AI Now, 12 2018.

[322] Wikipedia. "List of Cognitive Biases". URL https://en.wikipedia.org/wiki/List_of_cognitive_biases.

[323] Wikipedia contributors. List of languages by number of native speakers in india, 2022. URL https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India. [Online; accessed 6-May-2022].

[324] Will Dobbie, L. Andres, D. Paravisini, and V. Pathania. Measuring Bias in Consumer Lending. *Review of Economic Studies*, 88:2799–2832, 2021. doi: 10.1093/restud/rdaa078.

[325] B. Wilson, J. Hoffman, and J. Morgenstern. Predictive Inequity in Object Detection. 2 2019. URL http://arxiv.org/abs/1902.11097.

[326] J. N. Yan, Z. Gu, H. Lin, and J. M. Rzeszotarski. Silva: Interactively Assessing Machine Learning Fairness Using Causality. *Conference on Human Factors in Computing Systems - Proceedings*, 4 2020. doi: 10.1145/3313831.3376447.

[327] Z. Yan, P. Zhang, and A. V. Vasilakos. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42:120–134, 2014. ISSN 10958592. doi: 10.1016/j.jnca.2014.01.014.

[328] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K.-t. Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee. SUPERB: Speech processing Universal PERformance Benchmark. 2021. URL http://arxiv.org/abs/2105.01051.

[329] T. J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam. NetAdapt: Platform-aware neural network adaptation for mobile applications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11214 LNCS: 289–304, 2018. ISSN 16113349. doi: 10.1007/978-3-030-01249-6{_}18.

[330] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014. ISSN 23274662. doi: 10.1109/JIOT.2014.2306328.

[331] H. Zeinali, K. A. Lee, J. Alam, and L. Burget. Short-duration Speaker Verification (SdSV) Challenge 2021: the Challenge Evaluation Plan. Technical report, 2020. URL http://arxiv.org/abs/1912.06311.

[332] B. Zhang, N. Mor, J. Kolb, D. S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E. A. Lee, and J. Kubiatowicz. The Cloud is Not Enough: Saving IoT from the Cloud. In *7th USENIX Workshop on Hot Topics in Cloud Computing, HotCloud '15*, pages 21–21, Santa Barbara, CA, USA, 2015.

[333] Y. Zhang, T. Huang, and E. F. Bompard. Big data analytics in smart grids: a review. *Energy Informatics*, 1(1), 12 2018. doi: 10.1186/s42162-018-0007-5.

[334] Y. Zhang, N. Suda, L. Lai, and V. Chandra. Hello edge: Keyword spotting on microcontrollers. 2018. URL https://arxiv.org/abs/1711.07128.

[335] T. F. Zheng and L. Li. *Robustness-Related Issues in Speaker Recognition*. 2017. ISBN 978-981-10-3237-0.

[336] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang. Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019. doi: 10.1109/JPROC.2019.2918951.

[337] S. Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. Profile Books, 2019. ISBN 9781782832744.

[338] D. Zuehlke. SmartFactory - From vision to reality in factory technologies. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 17, 2008. ISBN 9783902661005. doi: 10.3182/20080706-5-KR-1001.4283.

List of Publications

2023

W. Hutiri, A. Y. Ding, F. Kawsar, and A. Mathur. Tiny, Always-on and Fragile: Bias Propagation through Design Choices in On-device Machine Learning Workflows. *ACM Transactions on Software Engineering and Methodology*, 4 2023. ISSN 1049-331X. doi: 10.1145/3591867

C. Rusti, A. Leschanowsky, C. Quinlan, M. Pnacek, L. Gorce, and W. Hutiri. Benchmark Dataset Dynamics, Bias and Privacy Challenges in Voice Biometrics Research. In *International Joint Conference on Biometrics (IJCB)*. IEEE, 2023

A. Gómez Ortega, J. Bourgeois, W. Hutiri, and G. Kortuem. Beyond Data Transactions: A Framework for Meaningfully Informed Data Donation. *AI & Society*, 2023. doi: 10.1007/s00146-023-01755-5

2022

W. Hutiri and A. Y. Ding. Bias in Automated Speaker Recognition. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, pages 230–247, Seoul, Republic of Korea, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533089 *** Awarded the TU Delft TBM ESS Best Paper Award in 2022 ***

W. Hutiri, L. Gorce, and A. Y. Ding. Design Guidelines for Inclusive Speaker Verification Evaluation Datasets. In *Interspeech 2022*, Incheon, Republic of Korea, 2022. International Speech Communication Association. doi: 10.21437/Interspeech.2022-10799

W. Hutiri and A. Yi Ding. Towards Trustworthy Edge Intelligence: Insights from Voice-Activated Services. In *2022 IEEE International Conference on Services Computing (SCC)*, pages 239–248. IEEE Computer Society, 2022. ISBN 978-1-6654-8146-5. doi: 10.1109/SCC55611.2022.00043

2021

W. Toussaint, A. Mathur, A. Y. Ding, and F. Kawsar. Characterising the Role of Pre-Processing Parameters in Audio-based Embedded Machine Learning. In *The 3rd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT '21)*, pages 439–445, Coimbra, Portugal, 2021. ACM. ISBN 9781450390972. doi: 10.1145/3485730.3493448

2020

W. Toussaint and A. Y. Ding. Machine Learning Systems in the IoT: Trustworthiness Trade-offs for Edge Intelligence. *2020 IEEE 2nd International Conference on Cognitive Machine Intelligence (CogMI 2020)*, pages 177–184, 2020. doi: 10.1109/CogMI50398.2020.00030

W. Toussaint, D. Van Veen, C. Irwin, Y. Nachmany, M. Barreiro-Perez, E. Díaz-Peláez, S. G. de Sousa, L. Millán, P. L. Sánchez, A. Sánchez-Puente, J. Sampedro-Gómez, P. I. Dorado-Díaz, and V. Vicente-Palacios. Design Considerations for High Impact, Automated Echocardiogram Analysis. In *International Conference on Machine Learning (ICML) 2020 ML for Global Health Workshop*, 2020. URL http://arxiv.org/abs/2006.06292

W. Toussaint and D. Moodley. Clustering Residential Electricity Consumption Data to Create Archetypes that Capture Household Behaviour in South Africa. *South African Computer Journal*, 32(2):1–34, 2020. ISSN 23137835. doi: 10.18489/SACJ.V32I2.845

W. Toussaint and D. Moodley. Identifying optimal clustering structures for residential energy consumption patterns using competency questions. In *SAICSIT '20: Conference of the South African Institute of Computer Scientists and Information Technologists 2020*, 2020. ISBN 9781450388474. doi: 10.1145/3410886.3410887

W. Toussaint and D. Moodley. Using competency questions to select optimal clustering structures for residential energy consumption patterns. In *International Conference on Learning Representations (ICLR) 2020 Workshop on Machine Learning in Real Life*, 2020. URL http://arxiv.org/abs/2006.00934

2019

W. Toussaint and D. Moodley. Comparison of clustering techniques for residential load profiles in South Africa. In *Proceedings of the South African Forum for AI Research*, Cape Town, 2019. URL http://ceur-ws.org/Vol-2540/FAIR2019_paper_55.pdf

Curriculum Vitæ

By bike, by foot, by truck and crowded bus on dusty road, from Arctic tundra to turquoise Turkish beaches, through Kyrgyz mountain ranges over Himalayan peaks, past Kashmiri pashmina vendors, through the Sahara desert to troubled Timbuktu, with a short stop in the bustling streets of London: on the 2 September 2019 Wiebke Hutiri (born Toussaint) finally arrived, on paths less travelled, from her birth place in Pretoria, South Africa, 11 849 days later, at the Technology Policy and Management (TPM) Faculty of TU Delft in the Netherlands to start her PhD.

Over a decade earlier, Wiebke's academic career commenced 10 000 kilometers due south at the University of Cape Town (UCT) in South Africa, where she completed a bachelors degree in mechanical engineering. She started her professional career in engineering consulting and then e-commerce, but returned to academia four years later to work as a data scientist at UCT's Energy Research Center (ERC). In this role, Wiebke's work in open data and data governance led her to publish the first large research dataset on domestic electricity consumption in South Africa - a historic and iconic dataset that captures a twenty-year longitudinal study that informed the country's post-Apartheid electrification programme.

Her work at the ERC inspired Wiebke to continue her studies, which led her to start a masters degree in computer science at UCT in 2016. In the years that followed, Wiebke completed her full research masters while continuing to work at the ERC. In her masters thesis, Wiebke studied approaches for using expert knowledge in the evaluation of machine learning systems to ensure their local and contextual relevance. She applied this work to create customer archetypes of South African residential electricity consumers. The masters degree was awarded to Wiebke with distinction.

At TU Delft, Wiebke has been able to deepen her expertise in the responsible design of socio-technical systems, with a focus on trustworthy artificial intelligence. Her work has been recognised and supported through numerous awards. In 2023 Wiebke received the TU Delft TPM Engineering Systems and Services department's best paper award. In the previous year, Wiebke received a grant from the Mozilla Technology Fund to support her work on fairness in voice technologies. Wiebke has also received a travel grant from the IEEE Women in Engineering association, and was selected as a Heidelberg Laureate Forum Young Researcher (2023), a fellow of the Diverse Intelligences Summer Institute (2022), a Data Science for Social Good Fellow (2019), and as an Emerging Leader on the TechWomen program of the US Department of State (2018).

From smart phones to speakers and watches, Edge AI is deployed on billions of devices to process large volumes of personal data efficiently, privately and in real-time. While Edge AI applications are promising, many recent incidents of bias in AI systems caution that Edge AI too, may systematically discriminate against groups of people based on their gender, race, age, accent, nationality and other personal attributes. More so, as the physical restrictions of Edge AI, together with the complexity of its heterogeneous and decentralised operating environment pose trade-offs when deploying AI to the edge.

This thesis is motivated by the societal demand for trustworthy AI, by the propensity of AI systems to be biased, and consequently by the need to detect and mitigate bias in diverse Edge AI applications. To address this need, this thesis develops design patterns for detecting and mitigating bias in the development of Edge AI systems. The design patterns present a generalisable approach for capturing established practices to detect and mitigate bias in machine learning. They make this knowledge readily accessible to researchers and practitioners that develop Edge AI, but who have limited prior experience with detecting and mitigating bias.