Models for Learning Reverberant Environments

by
Constantinos Papayiannis
MEng ACGI

A thesis submitted in fulfillment of requirements for the degree of
Doctor of Philosophy
of
Imperial College London

Communications and Signal Processing
Department of Electrical and Electronic Engineering
Imperial College London
2019

Copyright Declaration

Statement of Originality

I declare that this thesis and the research to which it refers are the product of my own work, under the guidance and supervision of my thesis supervisor Dr Patrick A. Naylor. Any ideas and quotations from the work of other people are properly acknowledges using standard referencing practices, whether these are published or unpublished. This work and the material of this thesis have not been submitted for any degree at any other academic or professional institution.

Abstract

Reverberation is present in all real life enclosures. From our workplaces to our homes and even in places designed as auditoria, such as concert halls and theatres. We have learned to understand speech in the presence of reverberation and also to use it for aesthetics in music. This thesis investigates novel ways enabling machines to learn the properties of reverberant acoustic environments. Training machines to classify rooms based on the effect of reverberation requires the use of data recorded in the room. The typical data for such measurements is the Acoustic Impulse Response (AIR) between the speaker and the receiver as a Finite Impulse Response (FIR) filter. Its representation however is high-dimensional and the measurements are small in number, which limits the design and performance of deep learning algorithms. Understanding properties of the rooms relies on the analysis of reflections that compose the AIRs and the decay and absorption of the sound energy in the room. This thesis proposes novel methods for representing the early reflections, which are strong and sparse in nature and depend on the position of the source and the receiver. The resulting representation significantly reduces the coefficients needed to represent the AIR and can be combined with a stochastic model from the literature to also represent the late reflections. The use of Deep Neural Networks (DNNs) for the task of classifying rooms is investigated, which provides novel results in this field. The aforementioned issues related to AIRs are highlighted through the analysis. This leads to the proposal of a data augmentation method for the training of the classifiers based on Generative Adversarial Networks (GANs), which uses existing data to create artificial AIRs, as if they were measured in real rooms. The networks learn properties of the room in the space defined by the parameters of the low-dimensional representation that is proposed in this thesis.

Acknowledgment

I want to express my gratitude to all the people that helped me achieve the goals I have set at the beginning of my PhD. I have learned many skills, both as a scientist and as a person, and made progress which is impossible for anyone to achieve alone.

I warmly thank all my friends who have always been a source of joy and laughter. I especially want to thank my wife Ute, who has been a source of inspiration and motivation for completing this Thesis. The extend of my gratitude towards my family is beyond words. They continuously supported me in so many ways throughout the years of my studies and always had faith in me. Lastly, a big *thank-you* to my supervisor Patrick Naylor for his excellent guidance and his positive attitude, which has been so important. Many thanks to Christine Evers, my co-supervisor, who always offered new perspectives and helped me learn a lot.

Contents

List of Figures

List of Tables

Abbreviations

ACE Acoustic Characterization of Environments. A noisy reverberant speech corpus and IEEE challenge run by the SAP group at Imperial College

AIR Acoustic Impulse Response

AI Artificial Intelligence

ANN Artificial Neural Network

ARMA Autoregressive Moving Average

AR Autoregressive

ASR Automatic Speech Recognition

AWGN Additive White Gaussian Noise

BC Bayes Classifier

BGSS Between-Group Sum of Squared Distances

CART Classification and Regression Tree

CNN Convolutional Neural Network

CTC Connectionist Temporal Classification

DCASE Detection and Classification of Acoustic Scenes and Events

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

DNN Deep Neural Network

DRR Direct-to-Reverberant Ratio

ECOC Error-Correcting Output Codes

EDC Energy Decay Curve

FDDRR Frequency-Dependent DRR

FDRT Frequency-Dependent Reverberation Time (RT)

FFT Fast Fourier Transform

FF Feed Forward

FIR Finite Impulse Response. A filter whose output is a weighted sum of past input values and whose system function contains only zeros and no poles.

FSS Forwards Sequential Selection

GAN Generative Adversarial Network

GA Genetic Algorithm

GMM Gaussian Mixture Model. An approximation to an arbitrary probability density function that consists of a weighted sum of Gaussian distributions

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

HMM Hidden Markov Model

IIR Infinite Impulse Response. A filter whose output is a weighted sum of both past input and past output values and whose system function contains both poles and zeros.

ISO Intl. Organization for Standardization

LASSO Least Absolute Shrinkage and Selection Operator

LPC Linear Predictive Coding. An autoregressive model of speech production.

LSTM Long Short-Term Memory

LS Least Squares

LTI Linear Time Invariant

MAP Maximum *a posteriori*

MA Moving Average

MFCC Mel-frequency Cepstral Coefficient

MINC Materials in Context Database

MIP Mixed Integer Programming

MIT Massachusetts Institute of Technology

ML Maximum Likelihood

MNIST Modified National Institute of Standards and Technology

MOCK Multi-Objective Clustering with automatic K-determination

MSE Mean Square Error

NBC Naive Bayes Classifier

NPM Normalized Projection Misalignment

PCA Principal Components Analysis

PDF Probability Density Function

PI Predictor Importance

PSD Power Spectral Density

ReLU Rectified Linear Unit

RGB Red Green Blue

RNN Recurrent Neural Network

RT Reverberation Time

ReLU Rectified Linear Unit

SED Sound Event Detection

SGD Stochastic Gradient Descent

SOM Self-Organizing Map

SVM Support Vector Machine

TD Time Distributed

TIMIT TI-MIT

TI Texas Instruments, Inc.

ToA Time-of-Arrival

TVAR Time-varying Autoregression

VAE Variational Autoencoder

VGG Visual Geometry Group

VRC Variance Ratio Criterion

WER Word Error Rate

WER Word Error Rate

WGN White Gaussian Noise

WGSS Within-Group Sum of Squared Distances

kNN k-Nearest Neighbours

t-SNE t-Distributed Stochastic Neighbour Embedding

Part I

Front Matter

Chapter 1

Introduction

1.1 Motivation and aims

Studying a textbook on acoustics provides a plethora of equations, which can be applied to real-life acoustic environments and describe the physical process of acoustic reproduction in the enclosure. No matter how many numbers are presented to a human however, the experience of listening to music in that room cannot be precisely conveyed. Thousands of coefficients can be used to create AIRs and auralisations can be created through the use of specialised software, but the results can be very different from reality [1]. Creating a description that can be provided and make humans really understand how sound would sound in an unseen environment is very difficult and communicating properties of the effect in any kind of way is challenging.

The same kind of problem is encountered when teaching machines anything about the reverberation effect. The typical form of describing acoustic environments is the thousands of FIR taps which correspond to the AIR. As it is hard for a human to make sense of those taps and translate them into high-level properties of the room, the same applies to a machine. The motivation of this work is to make machines more capable of making sense of the auditory world by improving their understanding of the reverberation effect. The aim is to propose an alternative representation for reverberant acoustic environments, which allows for machines to better understand them. The focus is on the understanding

of the properties which are important in discriminating between environments.

Capitalising on the advancements in the field of generative model estimation and deep learning, this thesis explores how machines can learn to create artificial AIRs associated with specific rooms. These examples can be provided to deep learning classifiers to better understand the reverberation effect and push the performance of state-of-the-art classification. Borrowing from the words of Richard Feynman,

> "*What I cannot create, I do not understand.*"
>
> —Richard Feynman

this is an important step for this research, which can improve the performance of many applications beyond the realm of classification. The method of generating data is improved by the choice of the domain of representation, which allows humans to directly understand properties of the generated environments and evaluate the quality of the solutions.

1.2 Thesis contributions

1.2.1 Research statement

This thesis aims to propose methods for classifying reverberant acoustic environments and methods to represent them, which enable machines to effectively and efficiently learn properties of the reverberation effect. The representations are to be low-dimensional and their parameters are to be semantically meaningful.

1.3 Original contributions

This thesis offers the following original contributions to the field

- Proposes feature domains for the classification of acoustic environments, constructed using acoustic parameters. Provides novel results, which compare the discriminative properties of the parameters on a set of classification tasks, by considering a number of classifiers. (Chapters 5)

- Proposes methods for the training of generalisable DNNs for the task of end-to-end room classification. A DNN architecture is also proposed for the task. Provides novel results about the performance of state-of-the-art classifiers on the task of discriminating between different reverberant acoustic environments and as to the features learned by the classifiers. (Chapter 6)

- Proposes a novel method for estimating the parameters that describe the early reflections in an AIR and the excitation used to measure it. (Chapter 7)

- Proposes a novel method for estimating the frequency-dependent absorption by the surfaces of materials in a room, using an AIR, measured in the room. Proposes a novel method for improving the modelling of early reflections, using the knowledge about absorptions by the surface of materials in the room. (Chapter 8)

- Proposes a novel method for data augmentation for the training of DNN room classifiers based on GANs, which improves the generalisation of the trained classifiers. (Chapter 9)

1.3.1 Publications

Publications by the author during this PhD that contributed to this thesis:

- C. Papayiannis, C. Evers, and P. A. Naylor, "Discriminative feature domains for reverberant acoustic environments," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, Louisiana, USA, Mar. 2017, pp. 756–760

- C. Papayiannis, C. Evers, and P. A. Naylor, "Sparse Parametric Modeling of the Early Part of Acoustic Impulse Responses," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 708–712

- C. Papayiannis, C. Evers, and P. A. Naylor, "End-to-end discriminative models for the reverberation effect," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (to be submitted)*, 2019

- C. Papayiannis, C. Evers, and P. A. Naylor, "Data augmentation using GANs for the classification of reverberant room," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (to be submitted)*, 2019

Publications by the author during this PhD that are not discussed in this thesis:

- C. Papayiannis, J. Amoh, V. Rozgic, *et al.*, "Detecting Media Sound Presence in Acoustic Scenes," in *Proc. Conf. of Intl. Speech Commun. Assoc. (INTERSPEECH)*, Hyderabad, India, Sep. 2018, pp. 1363–1367

After the publication of the above, their Python code implementation will be made available in an open-source format to other researchers in the field at:

`https://github.com/papayiannis/reverberation_learning_python`

1.4 Outline

The outline of this thesis is as follows

- Chapter II offers an introduction to the theory of acoustics and the reverberation effect. The technical background related to machine learning and acoustic environment modelling methods used in this thesis is discussed. Finally, a literature review is given on the modelling of acoustic environments and on sound classification.

- Chapter 5 begins the investigation on how reverberant acoustic environments can be classified in terms of properties of rooms. Acoustic parameters typically used in the field of acoustics are used to formulate feature domains for classifiers. The discriminative effect of each acoustic parameter is compared and evaluated across different tasks.

- Chapter 6 presents how end-to-end room classification can be done using state-of-the-art DNN classifiers and proposes strategies for training the networks. An analysis of the link between the extracted embeddings by the DNNs and acoustic parameters is given.

- Chapter 7 provides a model for the early reflections in AIRs, which represents them in terms of their ToAs and scales. This provides a low-dimensional and sparse representation as an alternative to the FIR filter taps.

- Chapter 8 presents a material presence-detector that estimates the frequency-dependent absorption of sound energy due to the materials in the room. It improves the modelling method of Chapter 7. Experiments in this Chapter present the application of the estimated parameters in the representation of entire AIRs, by combining them with a stochastic model for the reverberant tail.

- Chapter 9 provides a method for generating artificial AIRs, as if they were measured in real rooms. The generation is done using GANs, which are trained using the low-dimensional representation of AIRs discussed in this Thesis. This leads to the proposal of a method for data augmentation, used in the training of DNN room classifiers.

Part II

Background

Introduction

In this Part, concepts of reverberation, room acoustics and machine learning methods are introduced. A literature review on methods for the representation of reverberant acoustic environments and their classification is given.

This Part is organised as follows. An overview of the reverberation effect is given in Chapter 2.1. A discussion is made on how it is observed by humans and how this perception is linked to the principles of room acoustics. Section 2.2 discusses the FIR and Autoregressive (AR) representations of the acoustic channel and the benefits and disadvantages of each choice. Acoustic parameters related to the reverberation effect and their link to the perceived quality of the room's acoustics are introduced in Section 2.3. Fundamental technical concepts of machine learning are given in Chapter 3, which introduces supervised and unsupervised learning and DNNs. In Chapter 4 a literature review is provided. It provides a review of recent methods on channel classification, with a focus on reverberant environments and methods for their description and analysis.

Chapter 2

Reverberation

2.1 Overview

The reverberation effect is present in our everyday listening experiences. Our living rooms, kitchens, cars and houses are all reverberant acoustic environments. In fact, creating anechoic rooms is a very difficult task [7]. The effect can be aesthetically pleasing, especially in the case of music as it provides the feeling of warmth [8]. However, it can cause severe performance degradation to speech processing applications such as ASR [9]. Certain acoustic environments can also affect the intelligibility of speech by hearing-impaired listeners [10].

The building blocks of the reverberation effect are the reflections of sound waves off surfaces. Sounds emitted in an enclosure are reflected off its boundaries and the surfaces of objects within it. Microphones placed within the enclosure therefore receive many attenuated and delayed copies of the emitted sounds, whose superposition produces the reverberation effect [10]. Microphones with an unobstructed line of sight to the source will also receive a direct path sound from it. Reflections that reach the microphone within 50 ms after the direct path sound tend to reinforce it and are perceived as a single sound, with this effect being referred to as the Haas Effect [8]. In most cases, this perceived fusion of the early reflections leads to the *colouration* of the sound which is perceived as a change in timbre. The residual energy arriving after this point contributes to late reverberation, which is responsible for the smearing of energy across time and gives the feeling of space

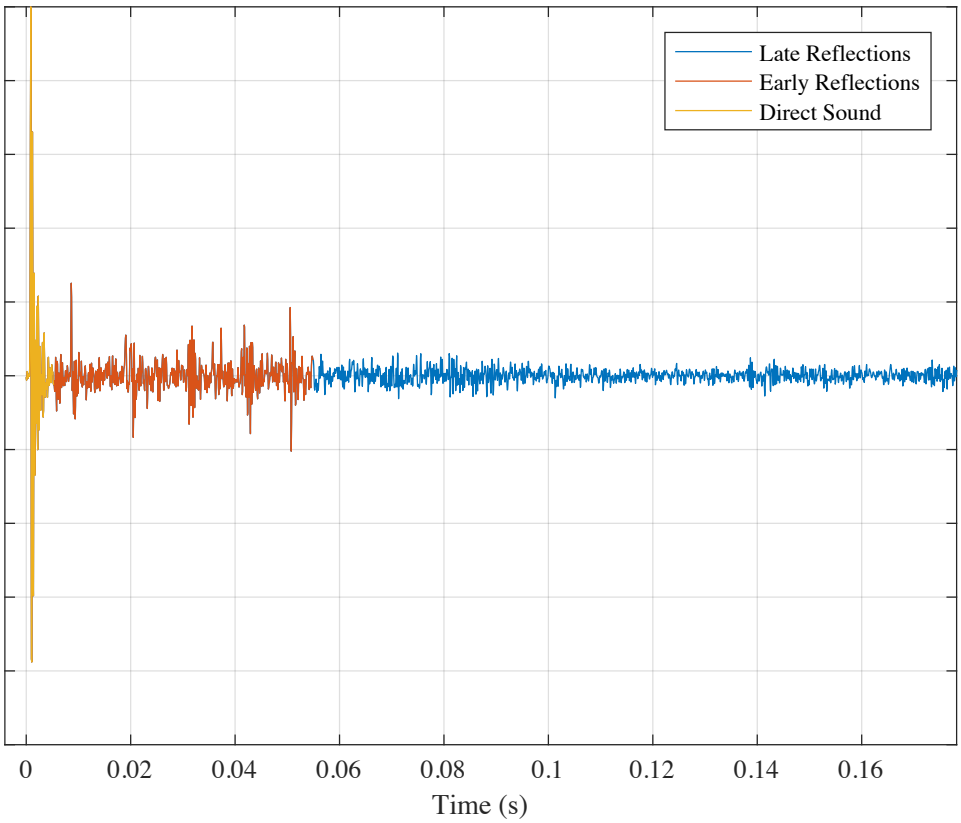


Figure 2.1: Early and late parts of an Acoustic Impulse Response (AIR) measured in a building lobby, part of the ACE challenge database [11].

[10]. To illustrate the different regions on an AIR waveform, Figure 2.1 is provided.

The paragraph above referred to a transitioning point in the reverberation process, between its early and late part. This transition point is referred to in the literature as the mixing time and it is associated with diffusion. When the acoustic energy is equally distributed in space and direction, the room is said to be diffuse [12]. Reflections in the enclosure can be specular and also contribute to the diffusion of sound energy. The material of the boundaries and factors such as the texture of the surface have a significant impact on the time it takes for sound energy in the room to be considered diffuse. The mixing time plays an important role in the analysis of reverberation [13].

Materials present in a room do not only affect the diffusion process but also define how fast sound energy decays. Different materials absorb sound energy differently to each other and a specific material absorbs sound energy differently at each frequency.

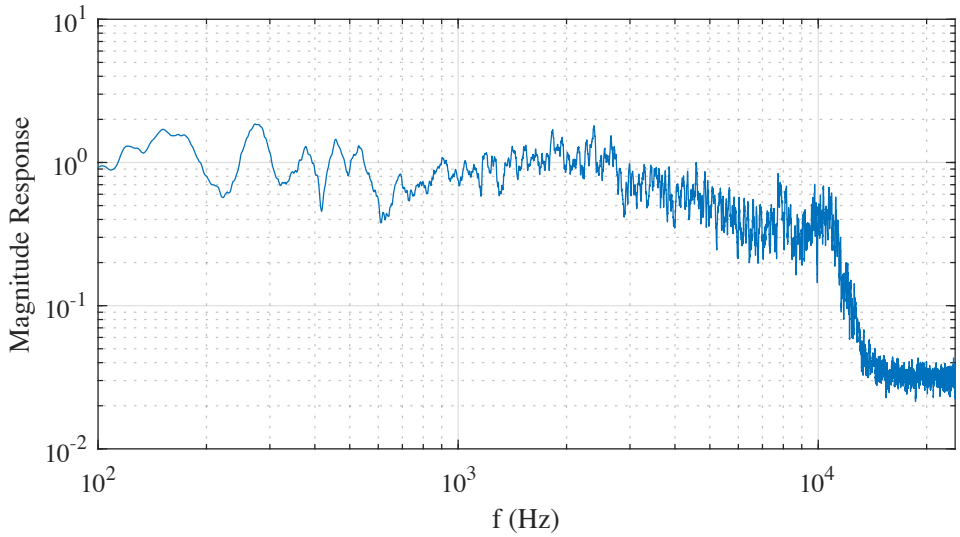


Figure 2.2: Spectrum of an AIR recorded in a building lobby, showing rooms modes more distinct in the lower part of the spectrum. AIR is taken from the ACE challenge database [11].

For instance, a room with open curtains and exposed glass windows will sound different from the same room with the curtains closed. This is because glass and fabric absorb sound very different to each other. Frequency-dependent absorptions by materials define the perceived spectral dynamics of the reverberant sound by causing sound energy in the room at different frequencies to decay faster or slower than others.

The effect of the material of the boundaries of a room has been highlighted but what is also very important is the relative position of the boundaries in the room to each other. This defines the room geometry, which in turn defines the room modes [7]. Room modes essentially define resonances at specific frequencies and their density at a certain frequency is proportional to the frequency value. Figure 2.2 shows the spectrum of a measured AIR in a building lobby. The Figure shows resonances as distinct peaks in the lower part of the spectrum and the opposite to be true for higher frequency regions. The region of the spectrum where room modes are resolvable is the region below the Schroeder frequency [10]. Above this frequency, the degree of overlap of the room modes is very high, which means that there is little point in evaluating any modes [7].

This basic discussion has introduced the effect of reverberation, consisting of many

reflections arriving at a microphone after the direct sound. These reflections are spectrally shaped by the materials in the room and cause resonances at frequencies defined by the room geometry. The study of the effect has led researchers to propose a number of parameters and mathematical models which are used to describe the effect and its perceptual qualities. The following Section reviews a small subset of models and parameters which are typically used to describe the effect.

2.2 Models for acoustic environments

The discussion about the effect of reverberation on speech in the literature often starts with a note on the contrast between its aesthetic qualities and its negative effects on machine understanding [10]. These contrasting facts have both been motivation for research in the literature. Methods have therefore been proposed for reverberating anechoic signals, in order to induce the aesthetic qualities of the effect, and also for dereverberating speech, in order to improve its understanding by machines. What the two tasks have in common is the use of models for the reverberation process. Finding compact and accurate models of AIRs has therefore been an active area of research for many years. A strong driving force for the work on addressing the high dimensionality of the models has been the dereverberation problem, which would in turn reduce the high computational complexity of existing approaches. This Section discusses the process of modelling an acoustic environment as either an FIR or an Infinite Impulse Response (IIR) filter. Other approaches involve the use of Kautz filters for modelling as in [14], [15]. Parametric forms of AIR have also been investigated for the purposes of artificial reverberation [16]. A review of the aforementioned models and others which are not discussed in this thesis is given in [10].

2.2.1 FIR filters

The reverberation effect in a reverberant enclosure is typically modelled as a convolution ($*$) of the anechoic signal $s(t)$ and the enclosure's AIR $h(n)$ as

$$\begin{aligned} x(n) &= h(n) * s(n) \\ &= \sum_{m=0}^{M-1} h(m)s(n-m), \end{aligned} \tag{2.1}$$

where n the sample index and M the total number of samples in $h(n)$.

The AIR is theoretically of infinite duration however in real measurements it reaches the ambient-noise floor in finite time. It is a representation of the acoustic channel in a room at the measurement source and receiver positions. Typically AIRs are given in the form of an FIR filter. For most real-life enclosures and at commonly used sampling rates, the filters involve thousands of coefficients. Due to its high dimensionality and nature, the AIR offers no directly interpretable information about the acoustics of the room and it is impractical to use for machine learning applications. It is however typical for acoustic environment databases to be available in this format, such as the ACE challenge database, as established methods for measuring AIRs exist in the literature [17].

Measuring AIRs in a given room can be done using a sine sweep method [17]. Measuring AIRs is not always practical and the most common case in the processing of speech would be to have the reverberant speech samples available only. Methods have been proposed in the literature for the estimation of the AIR from a given reverberant speech segment. In [18] a method is proposed to find this as a Least Squares (LS) estimate. In [19], using cepstra operations and reconstruction from phase-only, estimates of the AIR are calculated for a two-receiver scenario. The process however is very computationally expensive and makes a number of assumptions. In [20], the feasibility of the inversion is discussed for real-life acoustic scenarios. The work in [21] also focuses on practical issues around the invertibility of an AIR and presents choices that might prove problematic during the inversion stage.

2.2.2 IIR filters

Early work on the effect of reverberation explored alternative representations of acoustic environments to that of the FIR filter. In [22], [23], this was done by investigating the distribution of poles and zeros in reverberant rooms. Their distributions showed that poles tend to be invariant to the source and receiver locations while the contrary is true for the zeros [23]. The poles and zeros create coefficients of IIR filters, able to represent resonances and absorptions in a room. The work in [24] showed how using this representation, a smaller set of coefficients can represent an AIR when compared to an FIR filter. This investigation was extended in [14], [25] where the Frequency-Zooming Autoregressive Moving Average (ARMA) model was used.

IIR filters can model poles and zeros but can also have an all-pole form [26]. An all-pole IIR filter $V(z)$ of order P is described by coefficients a_p, which predict samples of a signal $s(n)$ using

$$s(n) = \sum_{p=1}^{P} a_p s(n-p). \tag{2.2}$$

The IIR modelling and all-pole modelling of room acoustics are investigated in this thesis as they are related to the resonances in a room, discussed in Section 2.1. Figure 2.3 shows how an all-pole model with 20 coefficients can represent the acoustic response in a building lobby. Using a small number of coefficients, the AR model captures the overall spectral envelope of the response. The frequency and magnitude of the poles are also shown in the Figure. Later parts of this thesis study the detection and modelling of these resonances for the purposes of classifying rooms and also for creating low-dimensional models for the reverberation effect. For the estimation of the coefficients, Autocorrelation LPC is typically used.

The coefficients can be transformed into cepstral coefficients [27], as this form has been shown to be more suitable for unsupervised learning applications [28]. Constraining the number of cepstral coefficients to be equal to P and simplifying the method of [28]

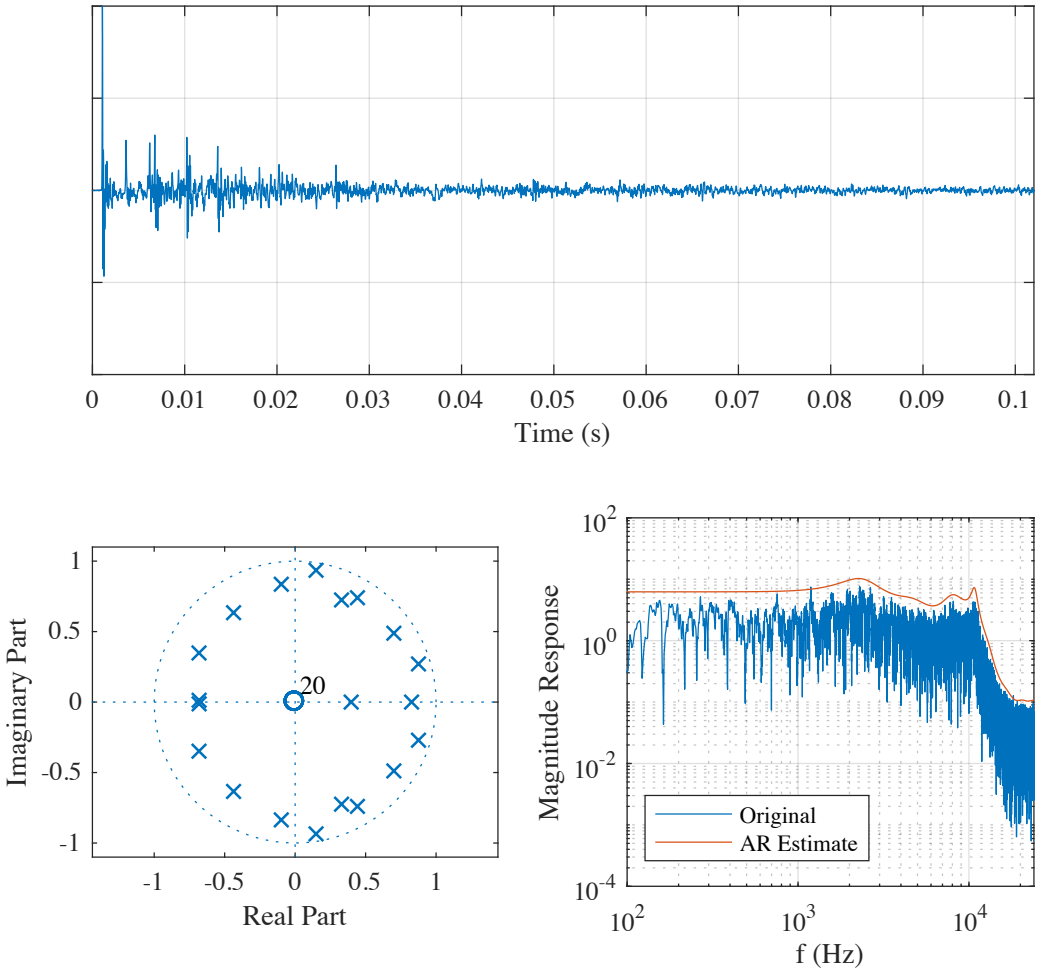


Figure 2.3: AR modelling of an AIR from the ACE challenge database [11]. Top: AIR as an FIR filter, Bottom left: 20 poles extracted through autocorrelation LPC. Bottom right: Spectrum reconstruction using all-pole AR model compared to spectrum extracted from the original AIR using DFT.

gives the following expression for the coefficient extraction

$$c_p = \begin{cases} -\alpha_1, & \text{if } p = 1 \\ -\alpha_p - \sum_{m=1}^{p-1} (1 - \frac{m}{p}) \alpha_m c_{p-m}, & \text{otherwise} \end{cases} . \tag{2.3}$$

The cepstral coefficients have properties in the Euclidean space which are advantageous for classifying environments [28].

The IIR filter representation is an alternative to the FIR representation of the AIR. An estimation of the order of the model, prior to the model fitting process, is however needed [26]. The invariance of the all-pole representation to source-receiver positions [22] is a desirable property for tasks such as room classification. This property also proves beneficial when aiming to estimate the parameters from reverberant speech. In [29], it was used to estimate the channel by modelling the source as a Time-varying Autoregression (TVAR) process and considering a time-varying all-pole filter.

2.3 Acoustic parameters for reverberant environments

Describing the acoustics in a room as dry, heavily reverberant or boxy is something humans are likely to do when asked to express their opinion about its acoustics. These terms are however subjective opinions of the individuals in question and although they provide a degree of understanding amongst us, they are not precise as to what they define. In order to have quantitative and standard descriptors about the acoustics of a room, parameter sets are used to describe them. These descriptions allow the transfer of understanding about an acoustic environment and also allow for ways to recreate the acoustics of it. A small subset of typically used parameters are discussed in this Section, which are used in this thesis for the task of categorising acoustic environments and also for representing them.

2.3.1 Absorption coefficients

Returning back to the terms used by humans to describe acoustics, it is common to describe certain rooms, such as large cathedrals, as *very reverberant*. What do we mean though when we say that? The same characterization would not be given to a carpeted living room with curtains. It might though be given to a tiled bathroom. There are certain traits in the rooms which alter the physical process of reverberation in the ways we perceive it. One such trait is the absorption of the materials. This Section discusses the notation used in this thesis to represent the sound absorption process, which is a topic investigated in depth in Chapter 8.

Considering the case of a perfectly smooth surface, incident sound of frequency f will be reflected specularly, resulting in reduced energy and different phase [7]. The complex factor that represents this process is denoted as R and is given by

$$R(f) = |R(f)| \exp(i\chi), \tag{2.4}$$

with χ representing the phase difference between the reflected and incident sound. The energy of the incident sound absorbed at the surface is described by the absorption coefficient

$$\alpha(f) = 1 - |R(f)|^2. \tag{2.5}$$

This coefficient is dependent on the frequency of the incident sound, as energy at different frequencies is reflected and absorbed differently [7]. Furthermore, the angle of incidence of the sound wave also affects the energy absorption.

Specular reflections back into the enclosure are not the only way that sound energy is reflected back from the material after incidence. Diffuse reflections also take place when the surface of the material is not smooth. This scattering of sound energy takes place when irregularities on the surface of a material are large compared to the wavelength of the incident sound. In that case, sound is reflected in many directions based on the distribution of the irregularities present, causing diffusion of the sound energy [7]. The scattering coefficient is a term used to describe the amount of reflected energy that is

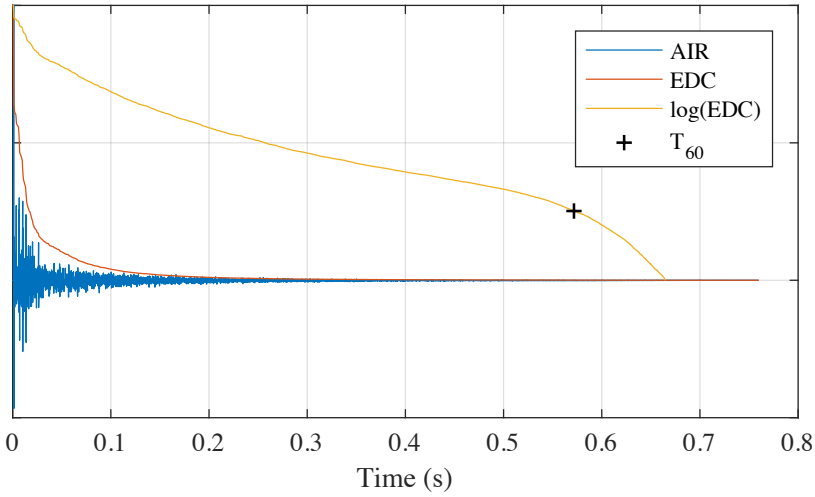


Figure 2.4: Measured AIR in a building lobby for the ACE challenge [11] and its corresponding Energy Decay Curve (EDC), with the T_{60} shown as the point where the curve decayed to -60 dB of its original level.

diffusely reflected from the surface [8]. In this thesis, the focus is on the modelling of specular reflections. The reflection coefficient R is considered to be a property of the material and not time varying. The effect of scattering and other residual effects such as the variance in the angle of incidence are accounted for separately. This is done through a single scaling process of the amplitude of reflections, which also models possible phase inversions.

2.3.2 Reverberation Time

Sound energy is injected into the room from a source which excites it. Once the sound source is switched off, the sound energy in the room decays in level. This decay is due to the absorptions, discussed in the above Section. Measuring the rate at which sound is absorbed, i.e. its decay, is done using the RT. The RT, which is also referred to as the T_{60}, is defined as the time it takes for the sound level to drop to -60dB with reference to its original level [7]. The curve which tracks the decay of the energy against time is called the Energy Decay Curve (EDC). It can be estimated from the AIR h using Schroeder

integration [30] as

$$\mathrm{EDC}(t) = \frac{\int_{\tau=t}^{\infty} h^2(\tau)dt}{\int_{\tau=0}^{\infty} h^2(\tau)dt}, \tag{2.6}$$

$$\mathrm{EDC}(t)_{\mathrm{dB}} = 10\log \frac{\int_{\tau=t}^{\infty} h^2(\tau)dt}{\int_{\tau=0}^{\infty} h^2(\tau)dt} \text{ dB}. \tag{2.7}$$

The EDC of an AIR measured in a building lobby is given in Figure 2.4, both in the linear and in the time domain, with the T_{60} denoted on the curve.

Polack has used the RT to model the reverberation effect in the diffuse field [31] considering an exponential decay of the energy of a White Gaussian Noise (WGN) process $\nu(t)$ [10] given as

$$h_{\mathrm{Polack}}(t) = \nu(t)\exp^{-\bar{\zeta}t}, \tag{2.8}$$

where the scalar

$$\bar{\zeta} = \frac{-3\log 10}{\mathrm{RT}} \tag{2.9}$$

is referred to as the average damping constant [10] and $\nu(t) \sim N(0, \sigma_\nu)$.

In work by Sabine [32], the reverberation time was shown to be inversely proportional to the sound absorption in the room and proportional to the volume of the room [10]. Recalling from Section 2.3.1, sound absorption from materials varies across the frequency spectrum. With this absorption being very influential to the sound decay in the room, the RT is very useful when described in its frequency-dependent form. Using Frequency-Dependent RTs (FDRTs) in [33], the parameters proved to be very informative for discriminating between different rooms. FDRT can be extracted using $^1/_3$ octave-bands and it is common to disregard the lowest part of the spectrum, which includes frequencies less the 150 Hz [33].

A method for estimating the RT of a room from the AIR was presented in [30]. The method involves nonlinear optimisation of model decay parameters after an AIR has been measured in the room. Measuring the RT from reverberant speech is possible, however challenges exist in getting reliable estimates [34].

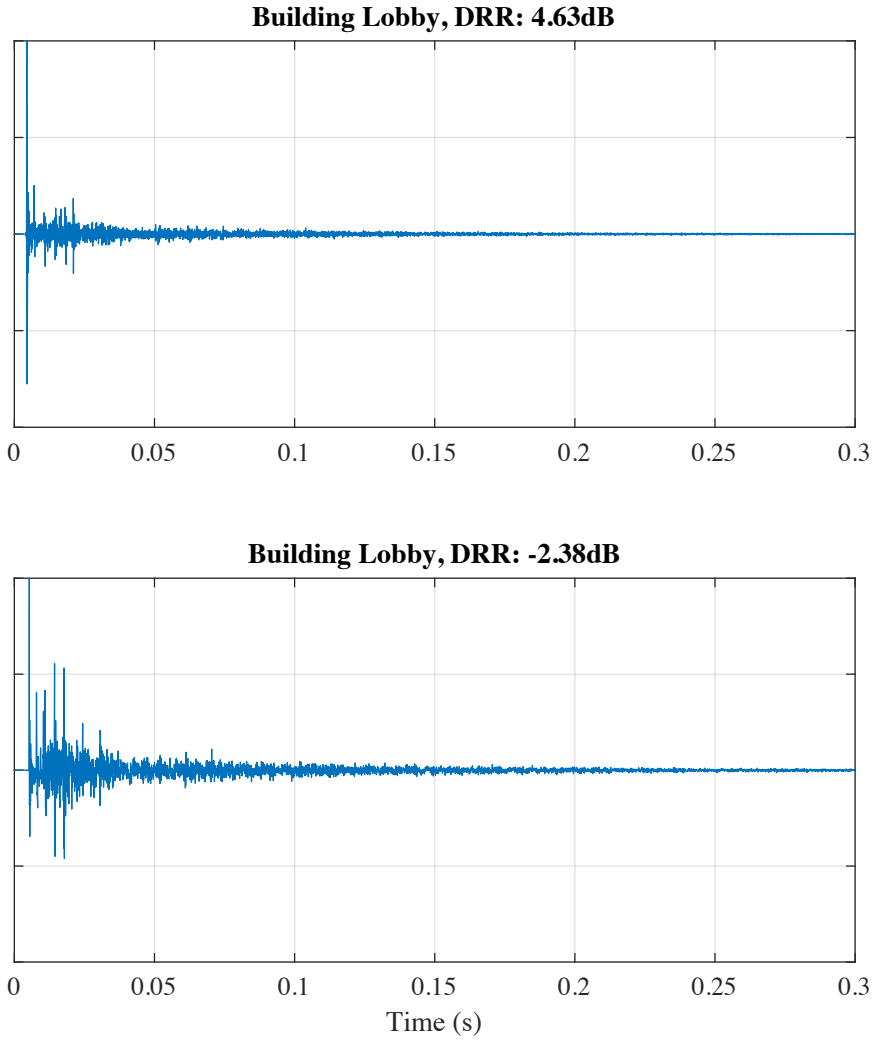


Figure 2.5: Estimated DRR from 2 AIRs measured in a building lobby for the ACE challenge [11]. High DRR (top) indicates a strong direct sound and low energy in the later reflections, with the opposite being true for the low DRR case (bottom).

2.3.3 Direct-to-Reverberant Ratio

In addition to the RT, another important objective measure of reverberation is the Direct-to-Reverberant Ratio (DRR) which is defined as

$$\mathrm{DRR} = 10\log_{10}\left(\frac{\sum_{n=0}^{n_d} h^2(n)}{\sum_{n=n_d+\Delta t}^{\infty} h^2(n)}\right)\mathrm{dB}, \tag{2.10}$$

where n_d the last sample that is part of the direct sound and h is the AIR. For the cases where the direct sound and reflections in the AIR overlap, (2.10) is an approximation. Explicitly modelling the direct sound as h_d and the rest of the AIR as h_r, leads to an accurate DRR estimation in the form

$$\mathrm{DRR} = 10\log_{10}\left(\frac{\sum_{n=0}^{\infty} h_d^2(n)}{\sum_{n=0}^{\infty} h_r^2(n)}\right)\mathrm{dB}. \tag{2.11}$$

The DRR effectively indicates the ratio of the direct sound energy over the reverberant sound energy. It is related to clarity [10], which is highly correlated to the performance of ASR systems [35]. The DRR is often perceived as an indicator of the proximity of the speaker to microphone, with low DRR values representing far-field conditions. Figure 2.5 shows two examples of AIR measurements, one with a high and one with low DRR. The high DRR system shows most of the energy concentrated in the direct sound and the opposite to be true for the case of low DRR.

To extract the frequency-dependent version of the DRR, similar to the RT case, $^1/_3$ octave-bands are used for filtering disregarding bands which include frequencies lower than 150 Hz.

2.3.4 Mel-frequency Cepstral Coefficients (MFCCs)

Both the RT and the DRR are parameters specifically designed to describe the reverberation effect. MFCCs, on the other, hand have been used extensively in the literature for ASR [36] and for classification of sound events and scenes [37]. Their historical use in classification tasks makes them an acoustic description for the channel to consider.

MFCCs provide spectral information in the Mel-frequency scale. The scale was experimentally derived based on the perception of frequency by humans [38]. The first step in extracting MFCCs is the design of a mel-filterbank which consists of triangular filters whose centre frequencies are equally spaced on the mel-frequency scale. After calculating the logarithm of the energy of the output of each filter, the coefficients are calculated as the Discrete Cosine Transform (DCT) of these terms. In terms of extracting MFCCs from AIRs in later Sections of this thesis, the entire response is used without segmentation into frames. For the case of speech, the aim is to extract the parameters that refer to the effect of the channel on the speech signal. One way of doing this for a long speech utterance is to segment the signal into frames, extract the MFCCs for each frame, then extract the average MFCC vector over all frames. This technique has been traditionally used to remove the effect of the channel from speech data prior to ASR [39]. However, in this case, this information can be exploited in order to learn more about the channel. The assumption is made that the channel would describe solely the acoustic environment. More information on the extraction of MFCCs and their applications can be found in [38]. An important point to note is that the coefficients carry information about the spectral shape of the signal being analysed. Furthermore, the DCT operation decorrelates the coefficients and concentrates the energy in a small number of coefficients, which reduces the dimensionality.

The MFCCs are used in this thesis to provide a description for the spectral shaping caused by the acoustic channel. The interest is in spectral regions where room acoustics play a significant role and also where speech energy is substantial. Therefore the frequency region of interest is limited to be between 100 Hz and 8 kHz, from which MFCC are extracted.

2.4 Summary

This Chapter has introduced room acoustics. Properties of the rooms that shape their characteristics in terms of acoustics have been described and acoustic parameters and models used to describe the effect have been presented. The next Chapter presents the

tools used in this thesis to enable machines to learn properties of the effect.

Chapter 3

Machine Learning

Tom Mitchell defines machine learning as the construction of "computer programs that automatically improve with experience" [40]. Its popularity in the scientific community has seen an exponential increase in the past years, which led to many exciting new technologies. It allowed for home voice assistants to be integrated into our households [41], photo-realistic synthetic videos to fool humans [42] and can only leave us to wonder about what the future holds for us from new innovations. The work done in this thesis embraces the breakthroughs in the field of Artificial Intelligence (AI) and brings room acoustics and reverberation into the focus. State-of-the-art methods based on machine learning are applied to tasks related to acoustics and are compared to classical approaches to solving a set of problems.

The discussion below introduces the fundamentals around state-of-the-art in machine learning. It aims to provide the reader with information on the technical tools used in the upcoming technical Chapters.

3.1 Artificial Neural Networks (ANNs)

3.1.1 Overview

When devising methods to enable machines to learn, a good starting place is to think about how we learn. ANNs stem from this motivation to understand how humans perform var-

ious cognitive functions. Stemming from a biological inspiration, the ANN nomenclature follows a similar principle. An ANN has neurons, which have corresponding output activations. These activations are a result of the activations of other neurons, which are connected to the input side with a corresponding set of weights. This interconnection of neurons resembles the brain connections, which create very powerful designs. In [43], it has been shown that with a sufficient number of neurons, networks can compute any function.

Learning using neural networks was introduced as rather intuitive in this Section, however it was not possible until 2006. It was not feasible to train deep networks until appropriate learning strategies were proposed [44], [45]. Creating deep networks by stacking ANN layers created powerful estimators and allowed for the interest in the field to significantly rise, with DNNs dominating the machine learning field in many scientific areas. The initial success of DNNs was first demonstrated in [44] using the MNIST dataset for handwritten character recognition, where DNNs surpassed the performance of the state-of-the-art at the time, SVMs. Later in 2012, Alexnet [46] managed to showcase the capabilities of DNNs and particularly CNNs for the task of image object-recognition. In the Image-net 2012 competition, Alexnet was the winner and has managed to surpass the second-best submission by a difference of more than 10% in the error.

The success of DNN models inspired researchers in many scientific areas to employ their use in respective fields. With regards to speech and audio processing, ASR has used DNNs to surpass the performance of Gaussian Mixture Model (GMM)-Hidden Markov Model (HMM) acoustic models by creating hybrid DNN-HMM models [47]. Other advancements in neural computation such as Connectionist Temporal Classification (CTC) loss layers [48] and Sequence-to-Sequence models [49] have allowed for end-to-end speech recognition [50], [51]. Sound Event Detection (SED) and Scene Classification have greatly benefited from DNNs. The Detection and Classification of Acoustic Scenes and Events (DCASE) challenge showcased how various architectures can be used for either of the two tasks [52]. The challenge has even demonstrated how techniques can be used to deal with the limited availability of training data and how deep learning can be done in those cases

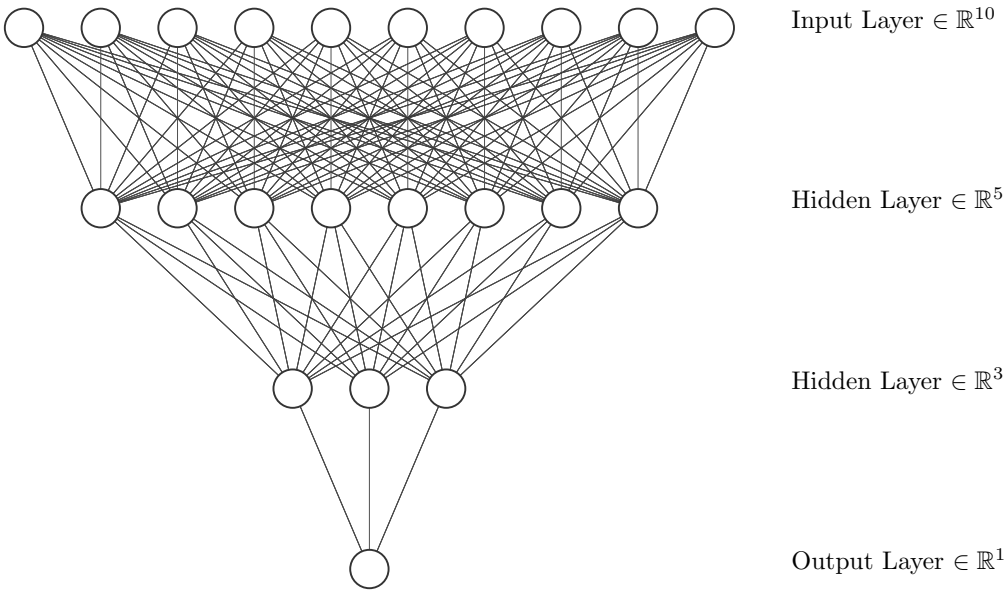


Figure 3.1: A Feed Forward (FF) network for binary classification with two hidden layers.

[53].

3.1.2 Deep Neural Networks (DNNs)

DNNs can approximate complicated nonlinear functions by stacking a number of layers hierarchically. The simplest DNN architecture is the FF architecture, which is illustrated in Figure 3.1. The network in the Figure employs 2 hidden layers for a binary classification task. For the mathematical expression of the process, the notation from [47] is followed. Neuron activations are defined using v and the vector of activations of all neurons in layer l is given by $\mathbf{v}^l$. Each neuron output z is passed through an activation function $f_a(\cdot)$. The biases at neurons of layer l are denoted as $\mathbf{b}^l$ and the connection weights from layer $l-1$ to layer l are denoted by $\mathbf{w}^l$. Therefore, the activations of layer l are given by the following vector $\forall\ l \in \{0, \ldots, L\}$

$$\mathbf{v}^l = f_a(\mathbf{z}^l) = f_a(\mathbf{w}^l \mathbf{v}^{l-1} + \mathbf{b}^l), \tag{3.1}$$

where L the number of layers on the network and $\mathbf{z}^l$ the neuron outputs of layer l, before passing through the activation function.

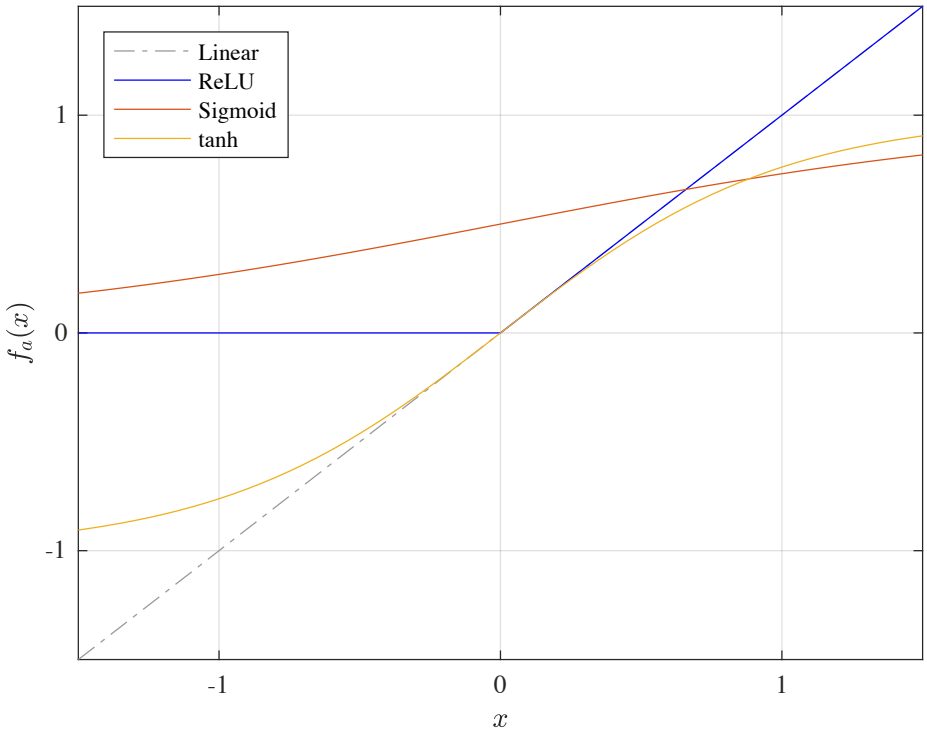


Figure 3.2: Neural network non-linear activation functions compared to linearity.

The choice of the activation function $f_a(\cdot)$ depends on the purpose of the network and task in hand. For the input and hidden layers, the sigmoid and tanh functions are typical choices for the activation functions $f_a(\cdot)$. The more modern Rectified Linear Unit (ReLU) function [54] is often used in the work done in this thesis as it leads to more effective and efficient implementations by switching off neurons entirely [47]. The resulting sparse activations have shown useful for discriminative tasks in [55]. Using ReLUs in the same work has led to better results than logistic networks in ASR experiments. A discussion on its benefits is given in [56], with a focus on image classification and sentiment analysis tasks. Figure 3.2 provides a visualisation of activation functions typically used for DNNs.

The activation function at the output layer is defined by the type of task performed by the network. For categorical classification, the activation function used is the softmax function. It provides the posterior probability of each class given the input vector, with the probabilities for all classes summing to 1. With each class represented by each output

neuron, the softmax function gives each activation as the following posterior [57]

$$p(c|\mathbf{x}) = \frac{\exp^{z_c^L}}{\sum_{i=1}^{C} \exp^{z_i^L}}, \tag{3.2}$$

where c the class index, and C is the number of possible classes. The use of the softmax function ensures that $\sum_{i=1}^{C} v_i^L = 1$. The classification of the input vector $\mathbf{x}$ is then done by the operation

$$\hat{y} = \underset{c}{\operatorname{argmax}}\, v_c^L. \tag{3.3}$$

What parametrises a given network is the set of weights, i.e. the connection weight between the neurons. When a network is designed, the weights have to be adjusted to best perform the desired task. This is what is referred to as model training. This is what Mitchell [40] defines as the process of machines to "improve with experience". For the training process, the following is required

- *Training examples.* These are given as pairs of observations and the ground truth for the output.

- *Loss function.* It is defined depending on the task in hand. For categorical classification, cross entropy is used as the loss function. The loss penalises confident and wrong decisions. It is defined as follows for an array of M training examples $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(M)}]^T$ with ground truths in vector $\mathbf{y}$

$$J = -\sum_{m=1}^{M} \sum_{c=1}^{C} q(y^{(m)}, c) \log p(c \mid \mathbf{x}^{(m)}) \tag{3.4}$$

$$q(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

- *Optimisation algorithm.* The algorithm collects the gradients using back-propagation [58] and follows an update strategy for the weights. Adam [59] is used extensively in the work done in this thesis for the training of the models. It incorporates desirable strategies such as momentum and the update of learning rates individually for each

trainable parameter.

- *Stopping rule.* Training, in theory, can continue infinitely, however this is of course not intended and a stopping criterion needs to be employed. Early stopping [47] can be chosen as the criterion. It monitors the loss function for training and validation data and makes a decision with regards to the termination of the process.

Learning steps are measured in epochs which typically refer to the number of times each training sample was used in the overall process. One epoch therefore refers to the use of each of the training examples once for a weight update step. Sets of samples are used to form batches, which are used to perform weight updates. The batch size is a hyper-parameter of the training algorithm which needs to be tuned for optimal performance. There exist practical limitations to the maximum batch size. Larger sizes might lead to faster convergence and more stable training but can however be too big to fit in the memory of Graphics Processing Units (GPUs), which are becoming popular for DNN training.

To train a network, the architecture needs to be defined. The architecture is defined by the engineer designing the network. Layers are designed, which are composed of neurons or cells of specific types and these layers are then connected together to form the entire network. The choice of the types of layers vary. The most typically used types of layers are discussed below.

3.1.3 Neural network layer types

The simplest and most fundamental ANN layer type is the Feed Forward (FF). This is the type of layers which compose the network of Figure 3.1. This and other common types of ANN layers are given below.

Feed Forward (FF)

These layers follow the concept introduced in Section 3.1.2. They output the weighted sum of the activations of the previous layer, which is subsequently passed through an activation function. Equation (3.1) is modified in such a way that it allows for the function $f_a^l(\cdot)$ to

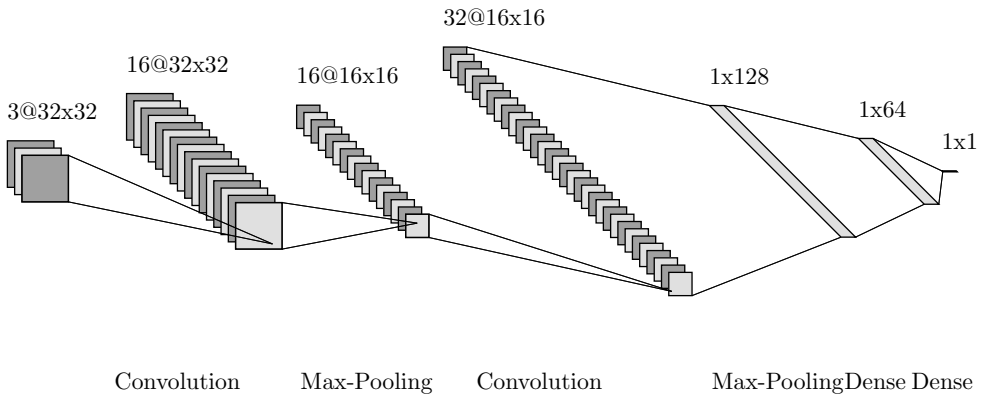


Figure 3.3: A Convolutional Neural Network (CNN) with two convolutional layers of 16 and 32 filters respectively, used for binary classification.

vary across layers and the resulting expression is the following

$$\mathbf{v}^l = f_a^l(\mathbf{z}^l) = f_a^l(\mathbf{w}^l \mathbf{v}^{l-1} + \mathbf{b}^l), \tag{3.6}$$

which provides the activations at the output of the l-th FF layer.

Convolutional

These layers employ the concept of filters on the inputs or neural activations [60]. The inputs are typically organised in channels. For example, in the case of Red Green Blue (RGB) images, three channels define the input layer. After the filters are applied to their input and the responses are passed through the relevant activations, the output is then of a number of channels equal to the number of filters. Max-Pooling layers [61] are typically applied to the output activations of convolutional layers which reduce their dimensionality while keeping the semantically important information. This way, convolutional layers can be stacked and form networks which are very deep, as the DNN proposed in [62].

The filters enable the network to detect higher level features in the image, with the activations of the filters commonly referred to as feature-maps. The filters aim to detect features in the image which are important to the task. These features can be edges in an image or specific gradients etc. With an image consisting of potentially millions of

pixels, convolutional layers allow for a small filter to be applied everywhere on the image to locate the presence of the high-level feature anywhere in the image. This is obviously much more economical in terms of trainable parameters than matrix multiplications, as it is the case of FF layers in (3.6). This is referred to as the *parameter sharing* property which allows the detection of *sparse interactions* by Goodfellow in [63]. In the same introductory material, the ability of the networks to detect *equivariant representations* is highlighted, as the filters are invariant to shift translations and brightness variations. Figure 3.3 shows a CNN configuration for binary classification.

Recurrent

These layers feature recurrent connections between a neuron's output and input. At the core of each layer are a number of cells which regulate the flow of information from the input and the current output to the future output of the neuron via gating. An example of very widely used recurrent cells are Long Short-Term Memory (LSTM) [64], with the focus now shifting towards the more modern Gated Recurrent Units (GRUs) [65]. They offer the ability to learn the long and short-term nature of the data, hence their name, and are extremely powerful for the case of sequence processing. They can be used instead of, or in combination with convolutional layers as investigated in [66]. Another advantage of using recurrent cells such as LSTM is that they can *trap* the errors in the cells during back-propagation, hence indirectly dealing with the exploding and vanishing gradient problem which can hinder the training processes [67].

Combining the above types of layers can provide with a wide range of architectures suitable for a variety of tasks, which is demonstrated in the technical Chapters.

3.1.4 Deep learning for generative model estimation

A generative model refers to a model of a process, with inputs $\mathbf{x}$ and $\mathbf{y}$, which gives the joint probability of $P(\mathbf{x}, \mathbf{y})$. On the other hand, discriminative models learn the conditional, or posterior probability, of $P(\mathbf{y}|\mathbf{x})$ [57]. In discriminative models, $\mathbf{x}$ is typically referred to as the features, although more recently the term is less often used given the

popularity of end-to-end models. $\mathbf{y}$ can either refer to a scalar class index or a binary vector for multi-task learning problems. The advancements in deep learning were highlighted by their performance in discriminative tasks [46] but also led to the recent proposals of Generative Adversarial Networks (GANs) [68] and Variational Autoencoders (VAEs) [69], which allow for the unsupervised estimation of generative models. GANs and VAEs differ in formulation but follow a very similar framework to reach their objective.

The task of estimating generative models was traditionally approached by estimating a parametric probability distribution $P(\mathbf{x}, \mathbf{y}; \theta)$, with θ the parameter set. This however is problematic for the case of distributions supported by low dimensional manifolds [70]. GANs and VAEs take a different approach to the task of estimating the generative model. Instead of estimating the parameters of a distribution, samples $\mathbf{z}$ are drawn from a known distribution $P(\mathbf{z})$ and passed through a set of parametric functions $f(\mathbf{z}; \theta)$, producing estimates of samples drawn from the underlying model. These parametric functions are modelled by DNNs and back-propagation can be used for their training.

In the literature review conducted for this thesis, GANs have appeared to be a more popular choice for generative model estimates. Variants of the *vanilla-flavour* GANs [68] have recently improved their initial drawbacks [70] making them suitable for training without having to adjust noise parameters using *ad hoc* procedures. Detailed discussions of their drawbacks and systematic ways of analysing them were given in [71] and [72], which allowed for principled training approaches to be proposed.

Generative Adversarial Networks (GANs)

GANs are composed of two networks which are posed as adversaries. The two networks play the roles of the generator and discriminator [68]. The task of the discriminator $D(\mathbf{y}; \theta_d)$ is to judge whether a given sample comes from the original data distribution or not. The task of the generator $G(\mathbf{y}|\mathbf{z}; \theta_g)$ on the other hand is to *fool* the discriminator into thinking that data samples it produces are originating from the original data distribution.

The two networks hence play the minimax game of

$$\min_G \max_D = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}} [\log D(\mathbf{y})] + \mathbb{E}_{\mathbf{z}\sim p_{\sim}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] , \tag{3.7}$$

which minimizes the Jensen–Shannon divergence between the generated data distribution and the true distribution, given an optimal discriminator.

The loss function (3.7), proposed with the original GAN paper in [68] can lead to unstable training, saturation of the gradients and mode collapse, which can be addressed by switching to the Wasserstein loss as proposed in [70]. This work applies GANs in novel ways to analyse acoustic environments. This first step in the understanding of the suitability of GANs for these applications therefore uses the original formulation of the loss in (3.7) for training. The exploration of the advantages of using alternative losses is not carried out in the relevant Sections of this thesis and it is reserved for future work.

3.2 Classification

The use of DNNs can lead to powerful estimators. One of the tasks that these estimators are used for is classification. The aim in a classification task is to answer the question: Which category does this belong to? *This*, refers to an input, which could be an image, an audio extract, text or any other data to be classified. The categories are defined by the problem in hand. For the case of object recognition, an example of categories would be images which contain horses and images which contain cats. Classification problems can be either multi-class or multi-task. In multi-class or categorical classification problems, each input sample can belong to only one class. In multi-task or detection problems, each sample can be positive for a number of classes. The horse and cat example posed above therefore would be a multi-task problem as images can contain both horses and cats. The task of classifying animals into horses, cats, dogs, etc., however is a multi-class problem as the classes are mutually exclusive. Classifiers are trained using given examples for each of the classes and learn to recognise recurrent patterns present in the input, characteristic of each class. This type of training, where data samples are labelled prior to training is

called supervised learning.

A broad range of classifiers exists in the literature. A set of classifiers is utilised in this thesis, a brief discussion on which is given below.

3.2.1 DNNs

DNNs can be used for the task of classification. Recalling from Section 3.1.2, using a softmax output layer with cross-entropy loss for training leads to a categorical classifier from a DNN. The classifier function is given by (3.3).

3.2.2 Bayes Classifier

Classifying inputs in terms of their classes can be done by using the posterior for each class each class as

$$\underset{i}{\operatorname{argmax}}\, p(c = i|\mathbf{x}), \tag{3.8}$$

where $i \in \{1, \ldots, C\}$. This is equivalent to how (3.3) classifies inputs in the ANN case. The ANN is designed to directly estimate this posterior, without explicitly modelling any part of the prior distribution of the input $\mathbf{x}$. This type of modelling is called discriminative modelling. Bayesian classifiers are generative models [57] that use Bayes theorem to transform the classifier to

$$\underset{i}{\operatorname{argmax}}\, p(\mathbf{x}|c = i)p(c = i). \tag{3.9}$$

The class priors $p(c)$ are estimated from the data. Known parametric forms can be used for the conditional Probability Density Functions (PDFs) $p(\mathbf{x}|c)$. Their parameters are estimated using a Maximum Likelihood (ML) or a Maximum *a posteriori* (MAP) [57] method.

With knowledge of the underlying statistics, the Bayes Classifier (BC) is an optimal classifier with respect to the misclassification error [57]. Estimating the parameters of the conditional PDFs $p(\mathbf{x}|c)$ can be very challenging. To simplify this problem, the Naive Bayes Classifier (NBC) makes the approximation that the input features are independent

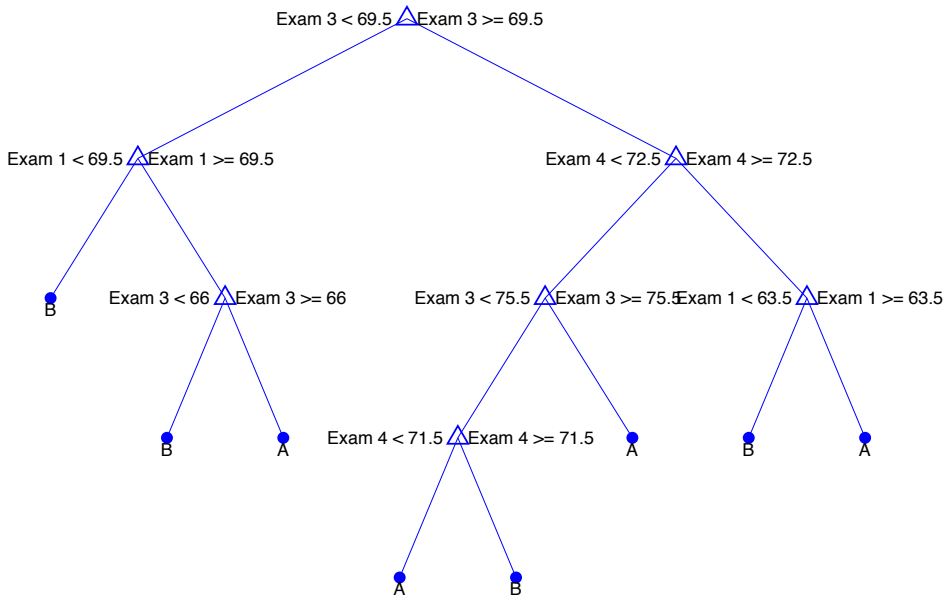


Figure 3.4: A CART grown to classify students in terms of their letter grade in a future exam. The results of 4 past exams of the student are used to estimate the class of the future grade. Each node of the tree makes a decision on the branch to follow depending on the result of the student in one of the past exams. The data for this figure is taken from the datasets provided with [75].

[73], which simplifies the classifier to

$$\underset{i}{\operatorname{argmax}}\, p(c=i) \prod_{x \in \mathbf{x}} p(x|c=i). \tag{3.10}$$

The suboptimal NBC can give better results than more complex classifiers such as SVMs when there is a shortage of data [74].

3.2.3 Classification and Regression Tree (CART)

CARTs take a different approach than the BC to the task of categorising an input. Instead of creating any PDFs, the model is composed of a sequence of if-then-else statements [76]. The root of the tree is one such statement, which checks whether a certain condition is true. The root of the tree then splits into two branches, one which is taken if the condition is true and one taken if the condition is false. The process is repeated on each branch. As

the input propagates through the nodes of the tree it eventually reaches a leaf. Each leaf in the tree corresponds to a class. Multiple leaves can correspond to the same class. The class is determined by the final leaf in the tree search.

Figure 3.4 shows an example of a CART, in this case trained to predict the letter grade of a future exam to be taken by a student, based on the student's grades in the past 4 exams. Each node asks a question regarding the grade of one of the past exams and the process is repeated until a leaf is reached. Each leaf indicates one of the possible letter grades. Naturally, doing a *dry-run* through the tree, one can see that if a student has achieved a perfect score in all exams, the predicted future grade is A. This thesis uses CARTs to categorise the acoustic environment using a set of estimated acoustic parameters, similarly to the above toy example.

3.2.4 k-Nearest Neighbours (kNN)

In the kNN framework [77], the classification does not depend on the estimation of the underlying model for the process in question but on the *neighbourhood* of a given input. In order to classify a new point, the k nearest neighbours to the input point from the training data are identified based on their distance. A majority-voting method is then used to classify the input based on the class membership of its neighbours. The majority-voting method dictates that the class membership of each neighbour will count as a vote and the class with the highest number of votes will be selected. The only parameter of the classifier is the choice of k. Setting k as not a multiple of the number of classes C avoids ties, which should otherwise be accounted for. The simplicity in the concept of the classifier means that no training is needed. The downside of this is that inference is computationally expensive. The k nearest neighbours need to be found from the training dataset, which turns its optimisation into a searching problem [78].

3.2.5 Support Vector Machine (SVM)

The deep learning renaissance [63] began when Hinton et al. managed to surpass the performance of SVMs in [44]. Prior to that, SVMs [79] were dominant in the field of

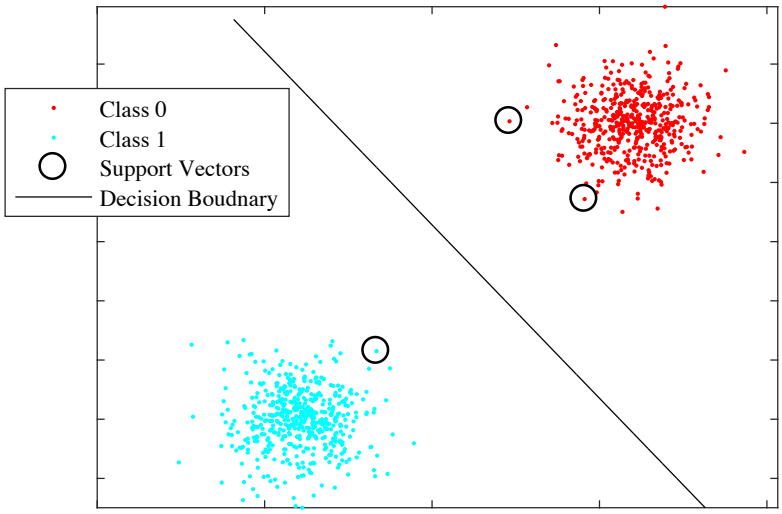


Figure 3.5: An SVM classifier drawing a maximum-margin discriminative hyperplane between two artificially generated datasets. The support vectors show the points from each class which support the hyperplane from each side. These are the points on each side which are closest to the boundary.

machine learning [57]. A SVM performs binary classification by learning a linear separator between two classes. It is characterised as a maximum margin classifier because its optimisation involves the maximisation of the margin between this discriminative line and the two closest points to it from each class in the training set. These points are named support vectors, which support the margin on each side. Figure 3.5 shows an example of an SVM trained on an artificial dataset with linearly separable classes. The support vectors and the decision boundary for the trained classifier are shown. During inference, the input is assigned to one of the classes by observing on which side of the boundary it lies on.

In real life, linear separation is not always feasible. The kernel trick [63] allows for nonlinear problems to be solved in a linear manner, which makes SVMs a very powerful tool for solving complex classification problems. They can also be extended to solve multi-class problems using one-vs-one or one-vs-all approaches [80] and using Error-Correcting Output Codes (ECOC) classifiers [81].

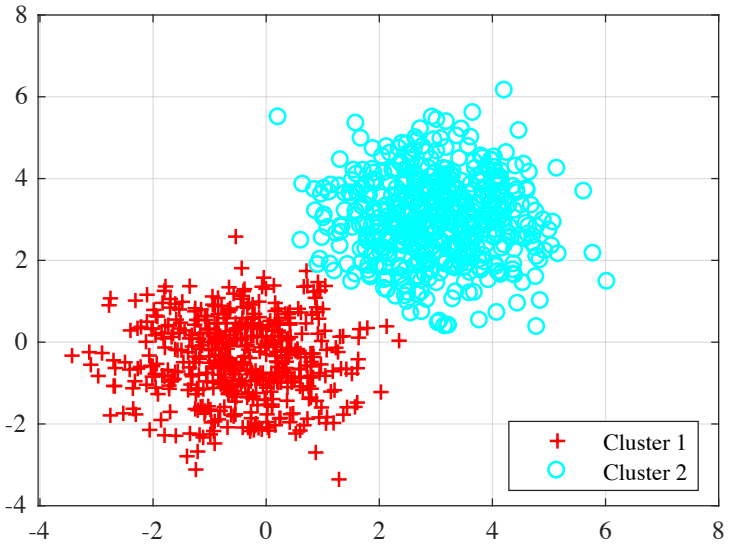


Figure 3.6: Clustering of artificially generated data. k-means clustering is used to separate the data into two clusters.

3.3 Cluster analysis

While classifiers can be very powerful, their supervised training relies on the availability of labelled data. This type of training, or learning, is referred to as supervised learning. In many cases, a large amount of data is available but without any labels as to their associated classes. Annotation of data is costly, both financially and in terms of time, as it often requires expert knowledge of the field. The process of learning from data directly and inferring properties about the distribution in the given space is called unsupervised learning. Cluster analysis is a subset of unsupervised learning and analyses the similarities and dissimilarities between samples in the training set [63], grouping them into groups called clusters. Clustering can unveil information about the underlying model for the observations in terms of its modalities and distribution in space.

The most widely known clustering algorithm is k-means [57], which assumes that the input space can be partitioned into Voronoi cells [57], each one corresponding to a class. The result of running k-means on an artificially generated set of points is shown in Figure 3.6. The points are automatically grouped into two clusters. The algorithm assumes that the number of clusters K is known prior to execution. The k-medoids [73]

clustering method, which is related to k-means, follows the same approach but with the centre of each cluster represented as one of its members. This allows for more flexibility on the distance measures used. k-means is a partitional clustering method [82]. More advanced clustering methods, such as Self-Organizing Map (SOM) [83] also fall under the same category. In SOMs, DNNs are configured in order to identify a Voronoi cell structure in the training data. These DNNs are trained using competitive learning, with each output unit competing to respond to the input.

Hierarchical clustering algorithms [73] produce a set of possible clustering solutions, in contrast to partitional clustering solutions, which produce only one. These solutions consist of the single cluster case, with all data belonging to the same group, and scale down to the case of each cluster containing one data point. Intermediate solutions are found by merging similar clusters together. The value in this process is the identification of clusters of data points and subsequent grouping of these clusters, at a higher hierarchical level.

Other approaches combine different clustering objectives for the analysis. Multi-Objective clustering using Genetic Algorithms (GAs) was proposed in [84], with the method named Multi-Objective Clustering with automatic K-determination (MOCK). It offers the ability to determine good clustering solutions in the space, by dynamically adjusting K. The optimisation is done by keeping dominant solutions, as typically done by evolutionary algorithms, based on the connectivity between points of the same cluster and intra-cluster variance. Furthermore, it can identify more complex structures in the data and not persist in the Voronoi cell assumption. The work in [82] discusses other similar methods for multi-objective clustering.

Evaluating the quality of a clustering solution serves as a tool to compare different solutions and select an optimal one, based on the evaluation criterion. A common task for this is the selection of the number of clusters K. In algorithms such as k-means, the number of clusters K is considered as known to the algorithm at initialisation. Therefore, the algorithm can be run for a set of candidate K values and the optimal solution according to the evaluation criterion can be used to determine the best value for K. The two following cluster evaluation measures are considered in this thesis.

- The Davies-Bouldin [85] measure is a cluster separation measure. This is defined [75] as

$$D_{k_1,k_2} = \frac{S_{k_1} + S_{k_2}}{Q_{k_1,k_2}}$$

$$\bar{R} = \frac{1}{K}\sum_{k_1=1}^{K} \max\{D_{k_1,k_2}\}_{k_2 \in \{1,\ldots,K\} \cap k_1 \neq k_2}, \tag{3.11}$$

  where S_k indicates the dispersion of cluster k and Q_{k_1,k_2} is the distance between the centroids of clusters k_1 and k_2. Smaller values of the measure indicate a better solution. The dispersion of a cluster is taken as the mean of the distance between all pairs of points in the cluster. Distances are Euclidean distances unless stated otherwise.

- The Variance Ratio Criterion (VRC) [86] or Calinski-Harabasz criterion [75], weights the Between-Group Sum of Squared Distances (BGSS) and Within-Group Sum of Squared Distances (WGSS) for M data points as

$$\text{VRC} = \frac{\frac{\text{BGSS}}{K-1}}{\frac{\text{WGSS}}{M-K}}. \tag{3.12}$$

  The BGSS and WGSS are defined as follows

$$\text{BGSS} = \sum_{k=1}^{K} M_i ||\bar{\mathbf{x}}_k - \bar{\mathbf{x}}||^2 \tag{3.13}$$

$$\text{WGSS} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathbf{X}_k} ||\mathbf{x} - \bar{\mathbf{x}}_k||^2, \tag{3.14}$$

  where $\mathbf{X}$ denotes the matrix holding the M data points to be clustered as the column vector collection $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]^T$. Similarly, $\mathbf{X}_k$ denotes the collection of the M_k data points which are members of cluster k. The vector $\bar{\mathbf{x}}$ denotes the mean of the vectors of $\mathbf{X}$ and $\bar{\mathbf{x}}_k$ the centroid of cluster k. Larger VRC values indicate a better solution.

3.4 Summary

This Chapter on the technical background of machine learning has provided an introduction to the field by presenting the concept of machines improving with experience. The discussion on classifiers presented methods which are used in later Sections to learn how to automatically classify inputs into classes. The discussion on clustering illustrated the concept of learning from data without labels. The next Chapter offers a literature review, which illustrates how machine learning and signal processing methods have been used in the literature to learn how to represent and classify reverberant environments before the contributions of this thesis are presented in the next Parts.

Chapter 4

Literature Review

This Section reviews existing work on the two areas which are covered in this thesis, namely the analysis and compact description of acoustic channels and their classification. The two topics have been studied in the literature separately, which justifies the separate review. One of the aims of this work is to bring advancements in the two fields together and improve the understanding of both tasks by doing so.

4.1 Acoustic environment analysis and description

4.1.1 Acoustic environment modelling

Alternative representations of acoustic channels to that of the typically used Moving Average (MA) process have long been a subject of interest in the literature. Early examples of this effort involve the use of ARMA models [24], motivated by their ability to efficiently model resonances in the room. Their use and extended methods for their estimation were revisited more recently in [25]. Their time-varying TVAR form was investigated in [87], where their estimation was made possible for the case of moving speakers. ARMA modelling leads to the construction of IIR filters, which were discussed in Section 2.2.2. Another type of filters, Kautz filters [14], were also recently studied for the modelling of the effect in [15]. As an alternative to the filter-based modelling, dimensionality reduction techniques have also been used as a direct search for the salient characteristics of

acoustic environments, operating on the taps of AIRs and using a range of techniques [88], [89]. The use of dimensionality-reduction techniques is effectively a search for the low-dimensional reverberation manifold, which can be used to reconstruct the effect. This diversity of approaches shows the interest in finding compact representations for the effect of reverberation, suitable for different applications.

Methods for the targeted modelling and simulation of specific parts of the AIR have also been proposed. Modelling of late reflections was typically proposed through the use of stochastic models that represent the reverberant tail. Variations of this technique have been used in the literature to combat practical issues in the rendering of AIRs, such as in [90]. The work in [13] specifically studies how the reverberant tail can be interchanged with a static tail model. The motivation was focused on reducing the computation complexity of artificial reverberation methods. The work in [91] approached the same problem and conducted experiments investigating the time where the transition between the early and the late parts is suitable. Both approaches were motivated by the perceptual effect of reverberation and have shown that after a time instance, referred to as the perceptual mixing time, replacing the tail samples with a stochastic model for the tail does not impact perceptual quality. A number of publications have approached the same research questions, investigating different parameters of the problem. A review is given in [13] and a discussion on theoretical aspects of the problem is given in [12]. In contrast to the samples corresponding to late reflections, which can be interchanged with samples generated by stochastic models, early reflections need to be accurately represented. The importance of this was illustrated in [92], which resorted to the use of samples directly from the AIR to represent the early part. The same stochastic modelling approach is not appropriate for the early reflections as they follow a structured distribution in time, related to room geometry and source and receiver positions [10].

The methods proposed in this thesis aim to capitalise on the benefits of existing approaches which model the late reverberation and the success of ARMA models. The focus of the work is on the modelling of the early reflections, for which novel methods are proposed. Stochastic and ARMA models are used for the modelling of late reflections. The

modelling methods are inspired by the physical process of acoustic reflections arriving at the receiver at specific ToAs. The surfaces of materials in the room will shape the spectrum of the reflections as they absorb sound energy differently across different frequencies. Modelling the reflection involves unravelling the properties which shape them, directly from their observations. Reconstructing properties of the room from observed reflections is called *inverse rendering*. The problem of inverse rendering received significant attention over the years and it is discussed below.

4.1.2 Inverse rendering

Inverse rendering methods vary in their focus, with each one targeting a different aspect of the environment. These aspects can be the dimensions of the room [93], the boundary location [94] or as in the case of this thesis, the materials present in the room.

Identifying the dimensions of a room was the focus of the work in [95], where multiple AIRs were used for 2D geometry estimation. ToAs were estimated and used as projective geometric constraints. A solution to the same problem was given in [96] and used information only from one AIR. The work in [97] focused on the fact that a number of assumptions are required by previous methods regarding the order of identified reflections. This led to a method which did not assume this *a priori* knowledge about the reflection orders. The problem was once more addressed in [93] but with the use of multiple receivers in order to estimate the room geometry. In [98], the focus was on the on 3D room geometry and the use of a single AIR with the problem posed as a least-squares optimisation. The localisation of boundaries in an enclosure was addressed in [94] with the process labelled *image-source locator*. The process relied on the estimation of the ToAs of reflections in the AIR and the clustered dynamic programming projected phase-slope algorithm is proposed for the estimation. It is interesting to note that most of the methods listed above rely on the estimation of ToAs of reflections. Reflection ToA estimation is studied in detail in this work and methods are proposed which allow their estimation directly from measured AIRs. The proposed methods are presented as an improvement to the existing ones in the sense that fewer assumptions are made prior to the ToA estimation.

Work done in the literature on the estimation of the material absorptions was mainly inspired by the field of architectural acoustics. Estimating the absorptions in the room allows for more accurate modelling, which in turn can make the analysis of the acoustics of a building much easier through simulation. Methods for the *in-situ* estimation of these parameters in [99] were based on the inverse boundary element method. These methods estimate the acoustic impedances of surfaces in a room, assuming that access to the room is available. In [100], a method was shown which is based on GAs that can infer the room shape and the wall absorption from an AIR alone. A single value was used to model the fullband absorption process from all walls. The issue of the non-uniqueness of the solutions was discussed, which indicates the inherent difficulty in the inverse rendering problem. The work in [101] assumed knowledge about the geometry of the room with the frequency dependent absorptions as the only unknowns. Using measured and simulated AIRs, aligned through optimisation of source-receiver locations, the material frequency dependent absorptions were tuned to match the two sets of responses. Scattering coefficients were taken into account at a later stage. GAs were used in [1] for the task and the proposed algorithms attempt to best match measured values for acoustic parameters such as RT and clarity to estimated ones. The work was revisited in [102], where the parameters to be matched and the optimisation approach were investigated further. The solution to the problem of finding the type of a material as well as its placement in space was transformed to a linear least-squares problem in [103], in order to get a specific AIR. To this diversity of methods proposed for the inverse rendering of acoustics, it is important to add methods used in computer-vision for solutions to a similar problem. Computer-vision methods have been used for the task of detecting the presence of certain types of materials in a room. The very successful method proposed in [104] involved a CNN model able to detect materials across 23 categories. Images captured allowed the patch-based and the overall material-recognition. The accompanying database to [104], called Materials in Context Database (MINC), was used in [105], where a CNN model was again employed to detect materials present. This initialised a model for the frequency dependent absorptions in the room. The method proposed in that work used this initial estimate to refine the frequency dependent absorptions iteratively to match measured AIRs. What can

be seen from the above is that computer-vision methods adopted state-of-the-art machine learning approaches for solutions to the problem, with audio-based methods lacking in their adoption. CNNs are however motivated by visual processing which makes their adoption in computer-vision a natural process. The work done in this thesis aims to bridge this gap and brings state-of-the-art machine learning in material-recognition and -detection from audio only.

Inverse rendering is used as a method to analyse the environment. This analysis helps construct a picture of high and low-level properties of the enclosure. These properties can be used to describe the reverberation effect to humans in a meaningful way, it can recreate the effect and it can also be used to infer similarities and dissimilarities between different rooms. The next part of this review looks at how similarities and dissimilarities between acoustic environments have been evaluated in the literature and how environments have been categorised into groups. The classification of audio inputs is considered in general, in order to illustrate how state-of-the-art machine learning has been used for audio inputs.

4.2 Classification of acoustic environments

The classification of an observed audio signal has been a fundamental concept embedded in the development of speech and audio processing algorithms from its roots. ASR and acoustic modelling traditionally began by classifying speech sounds into phonemes [39]. Other lines of research even consider how we, humans, make fundamental distinctions in the sounds we observe [106]. Therefore, distinctions and discriminative intentions form the motivation and aim for a number of applications around the processing of audio, helping us better understand and process our auditory world.

The tasks involving the classification of audio are segmented into the following three categories listed below and shown in Figure 4.1 for the purposes of this review.

- *Sound Classification* tasks involve the labelling of the received sound in terms of the event which generates the sound. This category includes SED and even ASR [37], [107] as the environment in which the sound or language atom is being produced

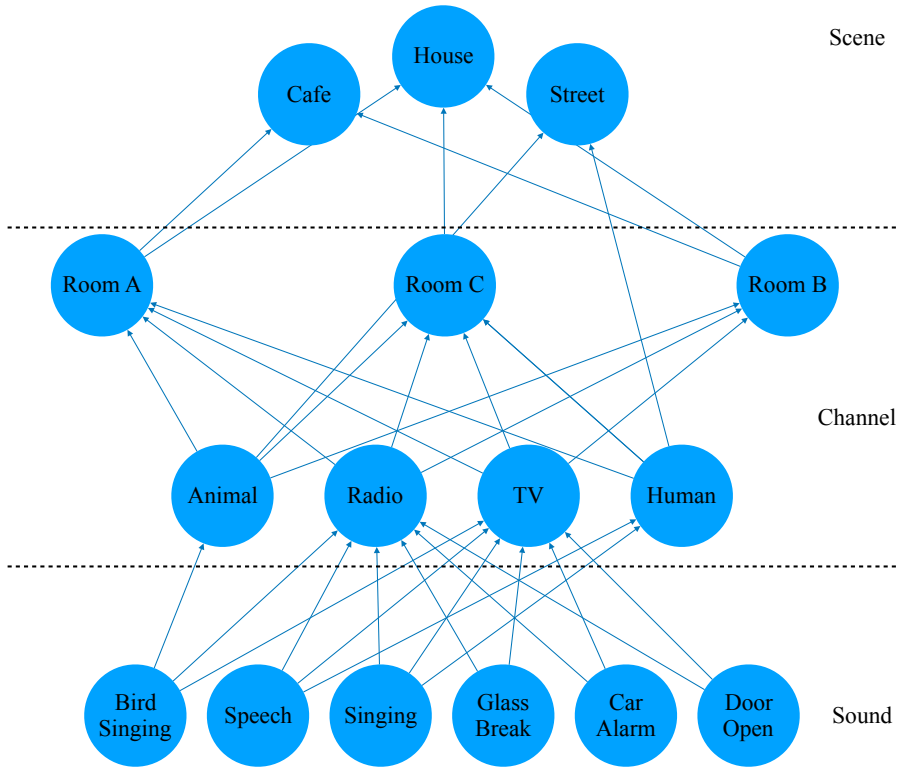


Figure 4.1: Classification of audio inputs illustrated hierarchically. Audio inputs can be classified in terms of the sounds contained in the audio segment or in terms of the channel that the sounds are transmitted through, including the room. The overall classification scheme identifies the scene, which encompasses all the above.

in is of no interest to the algorithm. These algorithms typically aim to remove the effect of the environment and transmission channel completely from the observation in order to operate robustly in a variety of scenarios. SED and scene analysis, which is discussed later, have been receiving increased research interest recently. A driving force behind this is the wider availability of sound data online, in databases and in data collections. This wider availability is due to our more widespread use of electronic devices such as home voice-assistants, content sharing through the internet and more content made available by entertainment providers. Millions of videos with sound, for instance, have been recently made available to researchers by Google [108], [109]. The DCASE challenge has given researchers the opportunity to showcase their achievements in the field and for the state-of-the-art to be available to the scientific community [110].

- *Channel Classification* tasks are involved with the classification of the environment in which sounds or speech has been recorded in. This can refer to the classification of the physical environment, such as room identification [2], [33]. To a finer level, the classification can be in terms of human-produced sounds and media playback [6] as this distinction is crucial for home voice-assistants. Channel classification also finds applications in forensics for the purposes of either tampering [111] or playback detection.

- *Scene Analysis* is the broadest category and aims to label the environment with a high-level descriptor [112], [113]. This descriptor is formed by the ensemble of the understanding of the sounds contained in an audio stream, the production medium and the room they are produced in. Listening to an audio stream for example which contains gunshots can lead to the understanding of a violent acoustic scene or a dangerous scenario. This label however would not be valid if the gunshots were played-back from a TV-set during an advertisement. Similarly, humans can attribute high-level labels to rooms such as *boxy*, simply by listening to an audio extract [10]. Perceptually derived understanding of the size and type of the room can also change the label of the scene, from an open space to a small living room, or a tiled bathroom. The above reasoning justifies the hierarchical representation of the tasks in Figure 4.1, with scene classification shown at the top.

Channel and environment classification is the category of interest in this work. The focus in on the classification of the room where the sounds are produced in. The classification would require an understanding of the effect of reverberation.

Classifying the reverberation effect has been addressed in the literature in the past. The motivation of some earlier work was to classify the effect in order to compensate for it through channel equalisation. In [114], one enclosure was considered with the aim to quantise the channel equalisation process for multiple sources and receiver locations. This was done through clustering using k-means on the all-pole approximations of the AIR, which later allowed for the construction of a codebook of equalisers. Another proposal for a solution to the same problem was given in [115], which utilised fuzzy clustering. This

method allowed for the contribution of various training examples towards the design of an equaliser that can be used in practice. In this proposal, the time domain representation of the AIR was used. Following from this work [116] proposed ways to reduce the computational complexity of such an approach.

The classification of the environment in terms of high-level characteristics of it was addressed later. Room classification was proposed in [117] through the use of MFCCs. These features were used to train GMMs for each room in the training set which are then used to form a ML classifier. A similar approach was later taken in [33] with FDRTs as the feature domain and a single Gaussian component used as the model. The concept of "roomprints" was introduced to reinforce the motivation behind the approach. Room classification was also addressed using decay statistics in [118].

4.3 Summary

This discussion concludes the introductory part of this thesis. A foundation on the reverberation effect was presented. The technical tools of machine learning which will be used in the following technical Chapters to understand it were illustrated. This Section offered a review of the literature on sound classification and reverberation modelling. The work presented in the later parts will start by showing how reverberant rooms can be classified into groups and the later Sections will use understanding of the effect to propose methods to improve this classification. This process provides a better understanding of acoustic environments which can help machines better understand acoustic scenes and therefore make better sense of their surroundings.

Part III

Discriminative Models for Reverberant Acoustic Environments

Introduction

Acoustic environments shape and define aspects of the sounds we hear and through this process, we experience the world around us from an audible perspective. At the same time, the environments provide listeners with cues that facilitate the understanding of their properties. Humans infer through sounds characteristics of the environment such as size. For example, as human listeners, we are able to tell whether we are sitting in a large concert hall compared to a tiled bathroom as the two would have very different acoustics. Discriminative models learned by machines can similarly distinguish between different types of acoustic environments [33]. Learning to perform these discriminations enables machines to better understand the world around them.

This Part explores the following two avenues for the design of discriminative models for acoustic environments:

1. Acoustic parameters are typically used to describe properties of the reverberation effect [10]. The first avenue explored in this Part investigates how effective is the use of these parameters as features for classification. This investigation is detailed in Chapter 5, which proposes feature domains for the classification of acoustic reverberant environments. The domains are proposed through the analysis of novel results, which compare the discriminative properties of a set of acoustic parameters across different classification tasks, by considering a number of classifiers.

2. Recent advancements in the field of deep learning improved the state-of-the-art in classification tasks [63]. A great benefit of DNNs is their ability to operate in an end-to-end fashion, hence not requiring the extraction of handpicked features. Chapter 6

proposes a method for the training of generalisable DNN classifiers, able to discriminate between reverberant rooms based on reverberant speech inputs. A DNN architecture is also proposed for the task. The investigation presented provides novel results with regards to the performance of DNNs on the task room classification and as to the features learned by the networks. This provides insight useful in a variety of tasks beyond room classification.

This two-sided analysis illustrates the advantages of using either a feature extraction based method versus an end-to-end model and experiments illustrate the cases where each one proves most beneficial.

Chapter 5

Discriminative Feature Domains for Acoustic Environments

This Chapter proposes feature domains that enable machines to learn how to accurately classify reverberant environments. The feature domains are formed by acoustic parameters, which are typically used to describe the reverberation effect. Novel results are given, which lead the proposal of the feature domains. The results are based on experiments that compare the discriminative properties of a set of acoustic parameters across a set of tasks. The environment classification tasks studied are room [117], room-type and room-position classification [114]. The task of selecting acoustic models for ASR [35] is also studied, in order to reduce the WER of decoding reverberant speech.

This Chapter's motivation is to gain insight into what are the observable properties of the reverberation effect that make categories of acoustic environments distinct. This investigation starts by training classifiers for acoustic environments using sets of acoustic parameters. The feature domains of the classifiers are kept to a low-dimensionality, as this increases the interpretability of the results. The proposed domains also find direct applications in the field and the low-dimensionality of the domains improves the computational and memory efficiency of relevant applications.

The set of acoustic parameters introduced in Section 2.3 is studied. All of the parameters have been used in the past to classify systems, as discussed in Section 4. Despite the

previous use of the parameters in the literature, there is a lack of a comparative study into their discriminative properties across a set of tasks, which is performed in this Chapter. The parameters can be estimated from reverberant speech, however these estimations will include an uncertainty due to non-idealities in the process. Analysis of estimation errors for specific methods is outside the scope of this work. Therefore, in order to provide a clear insight into their discriminative powers, parameters extracted from AIRs are used. Experiments are presented where artificial errors are introduced to the values of the acoustic parameters at controlled levels. This allows for an analytical comparison of the robustness of each proposed domain for classification.

The structure of the remainder of this Chapter is as follows. Section 5.1 provides the signal model and the classification framework used. Section 5.2 provides the feature selection process used. Section 5.3 provides the setup for the experiments and the investigation, which leads to the results of Section 5.4. The discussion in Section 5.5 provides the proposed domains and classifiers for the considered tasks. The resulting domains are later analysed in terms of their robustness. A conclusion is given in Section 5.5.

5.1 Classification of acoustic environments

With the motivation and aims for this Chapter outlined, this Section formulates the task and introduces the notation used.

5.1.1 Signal model

The reverberant speech signal $x(n)$ is modelled as a convolution process between the anechoic speech signal $s(n)$ and the AIR h. The tasks studied in this Chapter classify the input $x(n)$, based on properties of the environment in which h is measured in. In terms of notation, n represents the sample index and the additive noise is denoted as $\nu(n)$. In vector notation, the respective definitions are $\mathbf{s} = [s(0), \ldots, s(N-1)]^T$, $\mathbf{h} = [h(0), \ldots, h(L-1)]^T$ and $\boldsymbol{\nu} = [\nu(0), \ldots, \nu(N+L-1)]^T$. The convolution matrix $\mathbf{H}$ of dimensions $N + L - 1 \times N$ is constructed as

$$\mathbf{H} = \begin{bmatrix} h(0) & 0 & \dots & 0 \\ h(1) & h(0) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h(L-1) & \dots & \vdots & 0 \\ 0 & h(L-1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & h(L-1) \end{bmatrix}, \tag{5.1}$$

and the reverberant speech signal as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \boldsymbol{\nu}. \tag{5.2}$$

5.1.2 Acoustic features

The AIR is a description of the acoustic environment. It is measured by exciting the room using a sound source placed within its boundaries. Measuring the sound level at a position in the room using a microphone provides the AIR in the form of an FIR filter. Repeating this process M times during a data collection process provides M such AIRs. Moving the source or microphone or even changing the room, results in different AIRs. The rows of matrix $\mathbf{Y}$ are then formed by stacking the AIRs as

$$\mathbf{Y} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M]^T, \tag{5.3}$$

where the AIRs are zero-padded to match the length L of the longest one in the set.

Feature extraction operators are defined as $f_k(\mathbf{Y}) \; \forall \; k \in \{1, 2, \dots, K\}$, with

$$f_k : \mathbb{R}^L \rightarrow \mathbb{R}^{D_k} \tag{5.4}$$

and the transformation is summarised as

$$\mathbf{Y}_k = f_k(\mathbf{Y}). \tag{5.5}$$

The dimensions of the matrix $\mathbf{Y}_k$ are $M \times D_k$. The same relationship applies to vectors as $\mathbf{y}_k = f_k(\mathbf{y})$. As feature dimensions, the following acoustic parameters have been

Symbol	Dim.	Description
τ_f	1	The full-band Reverberation Time (RT).
τ_ξ	18	The RTs at the ξ-th sub-bands.
λ_f	1	The full-band Direct-to-Reverberant Ratio (DRR).
λ_ψ	18	The DRRs at the ψ-th sub-bands.
μ_ζ	25	The MFCCs at the ζ-th sub-bands.
κ_η	15	The η-th Autoregressive (AR) cepstral coefficients.
D	78	Total number of parameters considered.

Table 5.1: Notation used to represent the acoustic parameters used and their respective dimensionality.

studied that are linked to different aspects of reverberation: full-band RT [10], Frequency-Dependent RT (FDRT) [10], full-band DRR[1] [10], Frequency-Dependent DRR (FDDRR)[1] [10], MFCCs [38] and AR cepstral coefficients for the channel. In total $D = 78$ acoustic parameters are evaluated, listed in Table 5.1.

The acoustic parameters are used to form feature domains, which AIRs are transformed in via the use of the operators $f_k(\mathbf{Y})$. The interest is to find domains where discrimination between different environments is most effective in. The feature domains that are investigated in the following Sections are formed by combinations of the D considered parameters. The entire space of possible transformations is $K = 2^D - 1 = 2^{78} - 1 \approx 3.0 \times 10^{23}$. An exhaustive search of this space is not feasible, therefore this Chapter uses feature selection methods, able to identify which combinations of parameters lead to discriminative feature domains, without exhaustively testing every $k \in \{1, 2, \ldots, 2^D - 1\}$.

5.1.3 Classification framework

The previous Section has outlined the acoustic parameters that are considered in this Chapter. The aim is to evaluate the efficacy of each parameter for the task of classifying the acoustic environment. Before this evaluation, the notation used for describing the classification framework is defined in this Section.

Given M available acoustic environments, matrix $\mathbf{Y}$ is constructed and the feature

[1]For DRR estimation, the direct sound energy is taken as the portion spanning the range 0 to 2 ms after the arrival of the maximum energy sample.

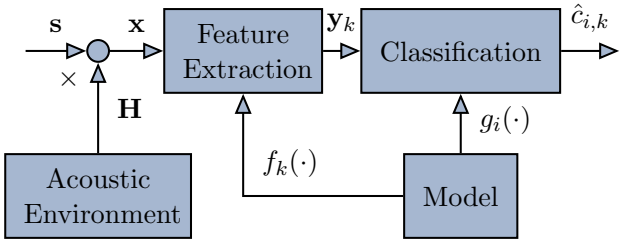


Figure 5.1: Diagram of the framework used for the classification of acoustic environments. The diagram shows the speech signal interacting with the acoustic environment. The resulting reverberant speech is then used by the model to classify the environment. The feature extraction operator transforms the speech signal to the corresponding feature space.

matrix $\mathbf{Y}_k$ is extracted, both composed of M rows. The vector of ground-truth values for the classes of the M AIRs in $\mathbf{Y}$ is $\mathbf{c} = [c_1, \ldots, c_M]^T$. Using this notation, the classifier g_i is defined as the function

$$\hat{\mathbf{c}} = g_i\left(\mathbf{Y}_k\right), \tag{5.6}$$

where $\hat{\mathbf{c}}$ denotes the prediction for $\mathbf{c}$ and the index $i \in \{1, \ldots, I\}$ indicates different classifiers. The process described so far is summarised in Figure 5.1.

The domain of $\mathbf{c}$ depends on the task and defines the possible classes for the classification problem. The four tasks considered are the following

1. Room-type identification, that discriminates between rooms of different types, i.e. offices, meeting rooms etc. This can be used to better understand the acoustic scene [119].
2. Room identification, that classifies the input in terms of the room where a recording was made in. This finds applications in forensics [33], [117], [120].
3. Room and position identification, where the room and the position where a recording was made in is identified. This finds applications in channel equalisation [115], [116], [121].
4. Acoustic model selection for ASR, which selects the acoustic model from a set of candidates, in order to minimise the WER [35].

Therefore $\mathbf{c}$ contains information about the type of room, the name of the room, the position within a room or an acoustic model in each case respectively. Throughout each of the given tasks, $\mathbf{c}$ remains fixed and the aim is to find a set of acoustic parameters to construct matrix $\mathbf{Y}_k$ that leads to the most accurate predictions of the class labels $\hat{\mathbf{c}} = g_i(\mathbf{Y}_k)$.

5.2 Feature domain construction

This Section presents the feature selection process that selects subsets of acoustic parameters that enable the classification of the acoustic environment. The discussion starts by presenting the acoustic environments studied, which are taken from a database of measured AIRs in a set of rooms.

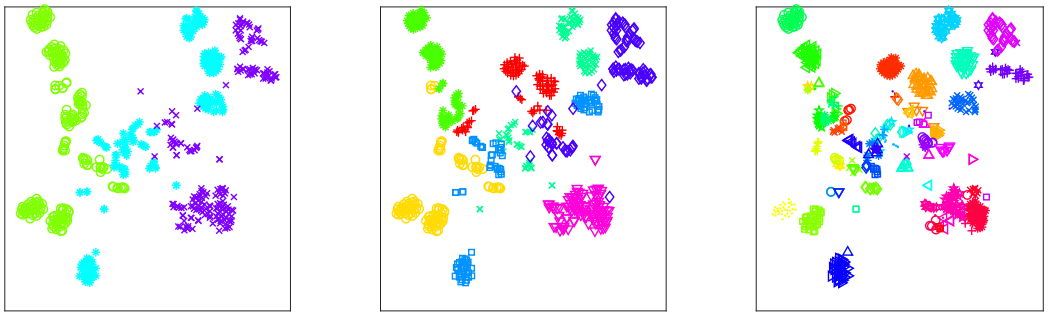
5.2.1 AIR database

For this work, AIRs provided for the ACE Challenge [122] are used. The database is composed of 70 AIR measurements involving 7 rooms and 3 different room types. Each measurement is multichannel and there is a variable number of channels in each. Considering each channel as a separate AIR, 700 AIRs are extracted and used. This information is represented in Table 5.2, which shows how the collected data is organised. The AIRs are processed in their original sampling rate of 48 kHz.

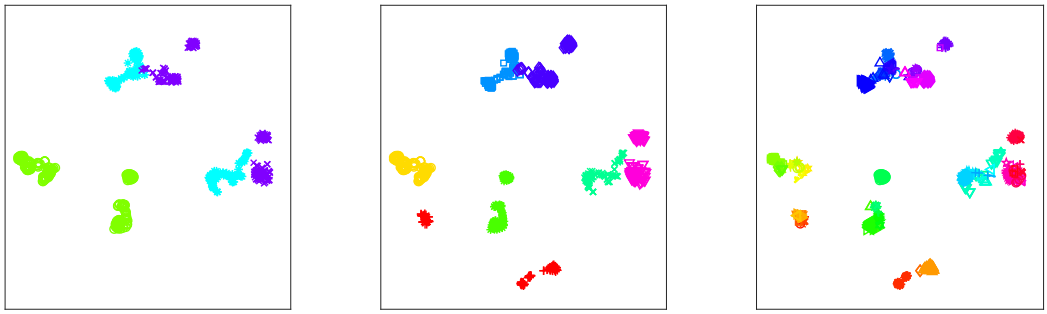
To visualise the data available in the database, the features of Table 5.1 are extracted from the AIRs and plotted with regards to each task in Figure 5.2. The dimensionality reduction method t-SNE [123] is used to visualise the distributions. Each column of plots represents one task, where different markers represent the type of the room, the room or the position within a room respectively for each of the three columns. Each row represents a different set of acoustic parameters.

Type	Room	Vol. (m^3)	Array	Measure. Positions	Num. Chann.	Measure. Positions	Num. Chann.
Meeting Room	Meeting Room 1	92	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
	Meeting Room 2	250	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
Office	Office 1	47	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
	Office 2	48	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
Lecture Room	Lecture Room 1	200	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
	Lecture Room 2	360	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
–	Building Lobby	72	Chromebook	2	2	10	100
			Mobile	2	3		
			Crucifix	2	5		
			Linear	2	8		
			EM32	2	32		
Total				70	700	70	700

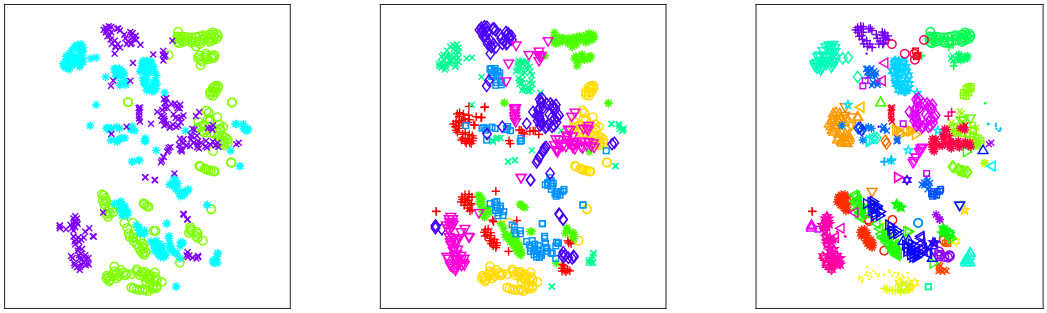
Table 5.2: Distribution of the data provided for the ACE database [34] in terms of the receiver-array, rooms and measurement position. The data shows the number of measurements made in each of the 7 rooms by each of the 5 receiver arrays used. The source and receiver positions are changed between each measurement for each array. The building lobby data is not used for the room-type experiments as it does not belong the same type as any other room.



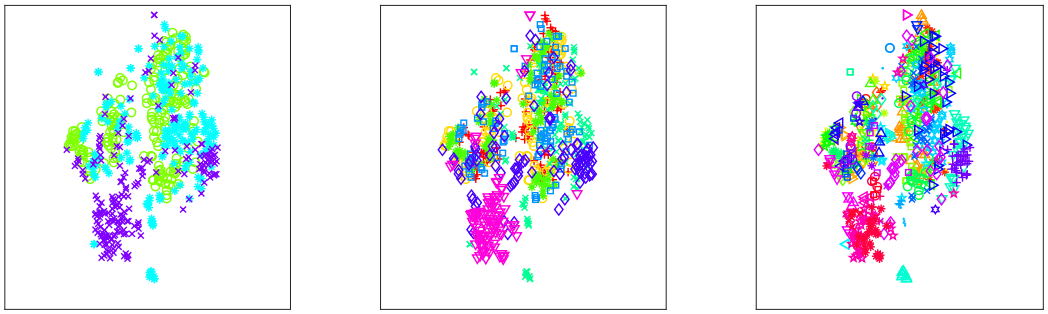
(a) Mel-frequency Cepstral Coefficient (MFCC)



(b) Frequency-Dependent RT (FDRT)



(c) Frequency-Dependent DRR (FDDRR)



Room Type Room Room & Position

(d) Autoregressive (AR) Cepstral Coefficients

Figure 5.2: Visualising the distribution of acoustic environments in t-SNE transformed feature spaces. Each column uses different markers for each environment to indicate different environment groupings. Left column: room types, middle column: rooms and right column: room and position.

5.2.2 Feature selection

For the construction of feature domains for the classification of acoustic environments, combinations of the acoustic parameters of Figure 5.2 are used. These combinations are used to train a set of classifiers of increasing complexity. The notation for the classification framework considered was introduced in Section 5.1.3. This involves the definition of the feature extraction function $f_k(\cdot)$, the classifier function $g_i(\cdot)$, the ground truth class for each feature vector c and its estimate $\hat{c}$. The aim is to find the best pair of feature extractor $f_k(\cdot)$ and classifier $g_i(\cdot)$ for each task. This is done by considering combinations of the candidate acoustic parameters and classifiers by evaluating

$$(k, i) = \underset{k,i}{\operatorname{argmin}} \; E_{k,i}, \tag{5.7}$$

where $E_{k,i}$ is the misclassification rate when using feature domain k and classifier i, given by

$$E_{k,i} = \frac{\|\mathbf{c} - \hat{\mathbf{c}}\|_0}{M}, \tag{5.8}$$

where $\|\cdot\|_0$ counts the number of non-zero elements of the vector.

The acoustic parameters considered were tabulated in Table 5.2. The classifiers considered were described in Section 3.2. They offer different separation mechanisms and assume different underlying model distributions. The considered classifiers are the following:

1. A NBC [57], using a Normal distribution to represent each class.
2. A NBC, using a GMM [124] to represent each class.
3. A CART [125]. The split criterion used is Gini's diversity index [76].
4. A kNN classifier, with $k = 3$ [57].
5. A SVM with a Gaussian kernel [124]. To allow for multi-class problems, an ECOC configuration [81] is used.
6. A DNN with 3 FF hidden layers, each one having a number of neurons equal to the number of input features. All activation functions are Rectified Linear Unit (ReLU)

functions, except the output nodes which use a softmax function [36]. The cost function used for training is cross-entropy, described by (3.4), minimised by the Adam optimiser [59]. The training is performed using batches of maximum size 50 and the networks are trained for 200 epochs. Input features are normalised to zero mean and unit variance.

Between the choices above, the minimisation search along k in (5.7) varies. This operation is effectively the feature selection process for each classifier. The classifiers considered are segmented into two groups, based on the way that feature selection is performed for each.

- For NBCs, kNN and for SVM, Forwards Sequential Selection (FSS) [126] is used. The number of candidate feature domains in this work is $K = 2^D = 2^{78}$. Rather than exhaustively searching this vast solution space, the FSS algorithm makes heuristic choices which make the problem tractable. For the purpose of FSS, the set of available AIRs is partitioned at the beginning of the feature selection process into a training and a test dataset, consisting of 85% and 15% of the data respectively. As FSS repeatedly trains and evaluates classifiers, the partitioning allows for this to happen. Stratified partitioning is used to ensure that the sets are representative of the overall population [75].
- For CART and DNN the feature selection is considered to be embedded in the algorithm. The two classifiers are able to disregard inputs that do not offer information that is useful in classifying the training examples.

The following Sections illustrate the process of training the classifiers above and the result of the feature selection process. The aim is to understand what is the best classification scheme for each task and to propose a feature domain for each one. This will provide insight into which low-level features of the acoustic environment define the high-level classes used to describe them.

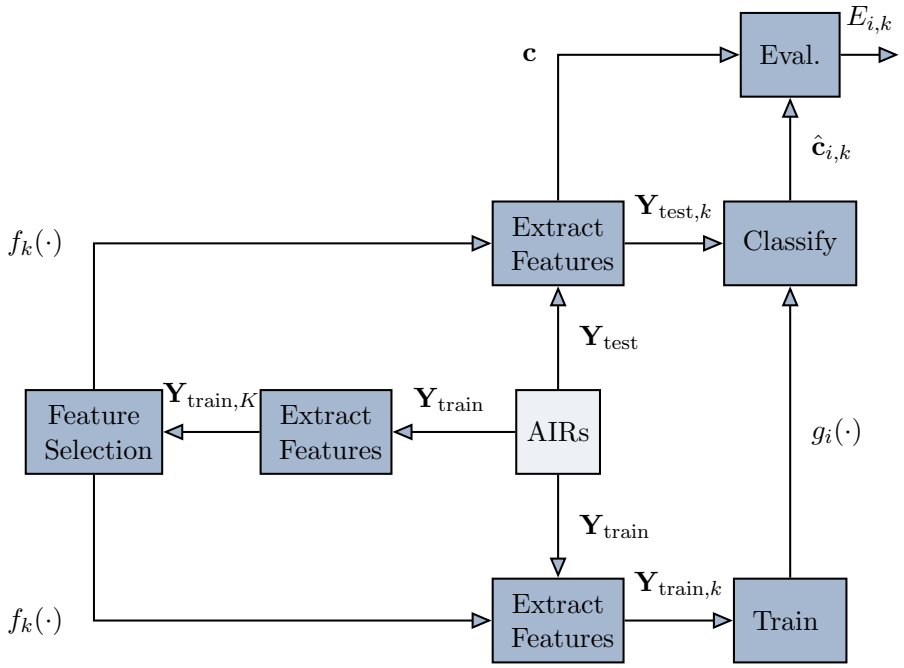


Figure 5.3: Diagram of the process of training a classifier, the feature selection and the evaluation of the resulting models for classifiers using Forwards Sequential Selection (FSS).

5.3 Experiment setup

A set of acoustic parameters and a set of classifiers are considered for discriminating between different acoustic environments. The following experiments and the subsequent analysis identify suitable feature domains and classification schemes for each of the tasks studied.

5.3.1 Setup and evaluation

The designed experiments consist of the following steps for each task and each classifier, which are illustrated in Figure 5.3.

1. *Feature extraction.* The 78 acoustic parameters described in Table 5.1 are extracted from the 700 AIRs of Section 5.2.1.

2. *Feature Selection.* A set of features is chosen as described in Section 5.2.2 for the task of separating the environments into different categories.

Task	Num. Classes	Num. Folds	AIRs per Fold	Folds
Room-type	3	6	100	Rooms.
Room	7	70	3–32	Positions.
Room-position	70	700	1	Individual AIRs.
ASR model	4	7	100	Rooms.

Table 5.3: Segmentation of the 700 AIRs provided by the ACE challenge [11] database into classes based on each task. The splitting of the data into folds for the evaluation of the trained classifiers is shown, with a description of what each fold represents.

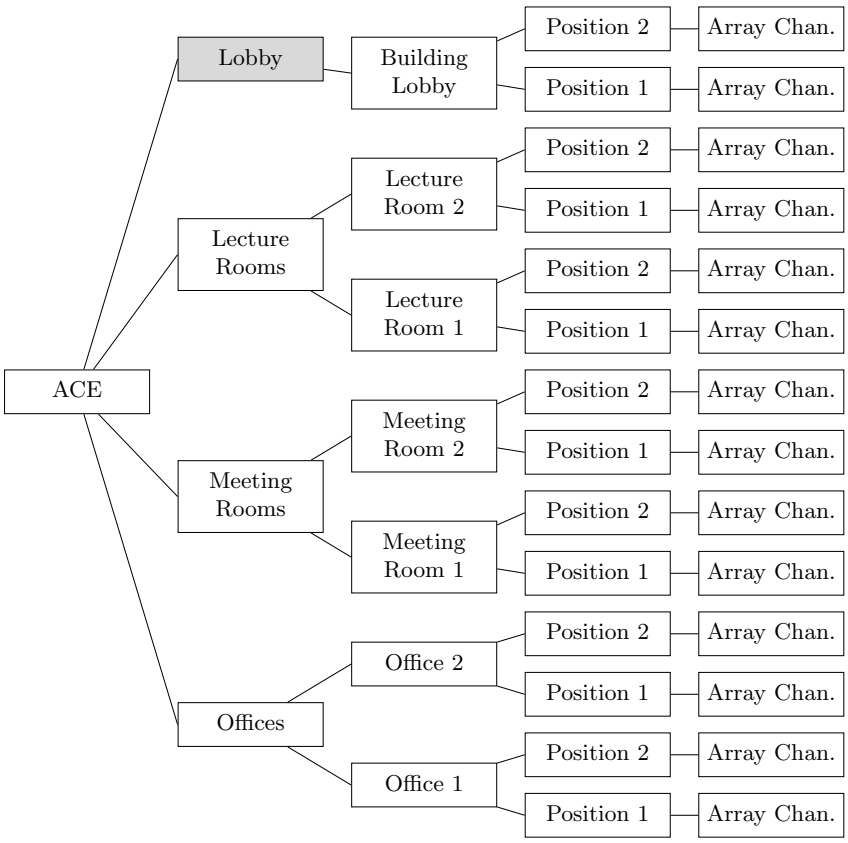


Figure 5.4: Structure of the ACE AIR data along with their roles in each classification task and in the cross-validation evaluation.

3. *Training.* The classifier is trained using the chosen features as the input.

4. *Evaluation*: The trained classifier is evaluated with test data.

Note that for CARTs and DNNs, steps 2 and 3 form a single step, as the feature selection is part of the training process.

The evaluation is done using the misclassification rate of (5.7) through cross-validation. The construction of the cross-validation folds is dependent on the task and aims to represent reasonable test scenarios for each one. The segmentation of the data into classes and folds for all tasks is shown in Table 5.3 and visualised in Figure 5.4. The feature selection process is run prior to cross-validation for one of these folds, which is randomly chosen. Recalling from Table 5.2, the number of channels per receiver varies between 3–32, hence the number of measurements per position varies and so does the number of AIRs per fold for the room classification task. For room-type classification, the type *Lobby* is not included as it only contains one room.

The classes of the ASR model selection task in Table 5.3 are indicated as a number of acoustic models. For this task, a classifier is given a reverberant environment and effectively has to choose which acoustic model to use to better understand the reverberant speech produced in the environment [35]. The objective is formally defined as choosing the model that minimises the Word Error Rate (WER) [36] in the decoding of reverberant speech [35]. Unlike the other considered tasks, for this one the ground-truth labels, i.e. which one is the best model to use for each of the 700 AIR, are not available from the database. To create these labels, the first step is to create the acoustic models for this task. For this purpose, 3 sets of 5 AIRs are randomly chosen from the database. These are used to train 3 acoustic models for ASR (one from each set). The chosen AIRs were measured in two meeting rooms and one building lobby. For the training of each acoustic model, the training speech corpus of TIMIT [127] is used and each utterance in the corpus is convolved with a randomly chosen AIR from the set of 5. An additional model is trained, using no AIRs but only the anechoic speech from the corpus, giving a total of 4 trained acoustic models. The remaining 685 AIRs are used individually as test environments. Each one is convolved with the entire test corpus and then decoded by each of the 4 acoustic models.

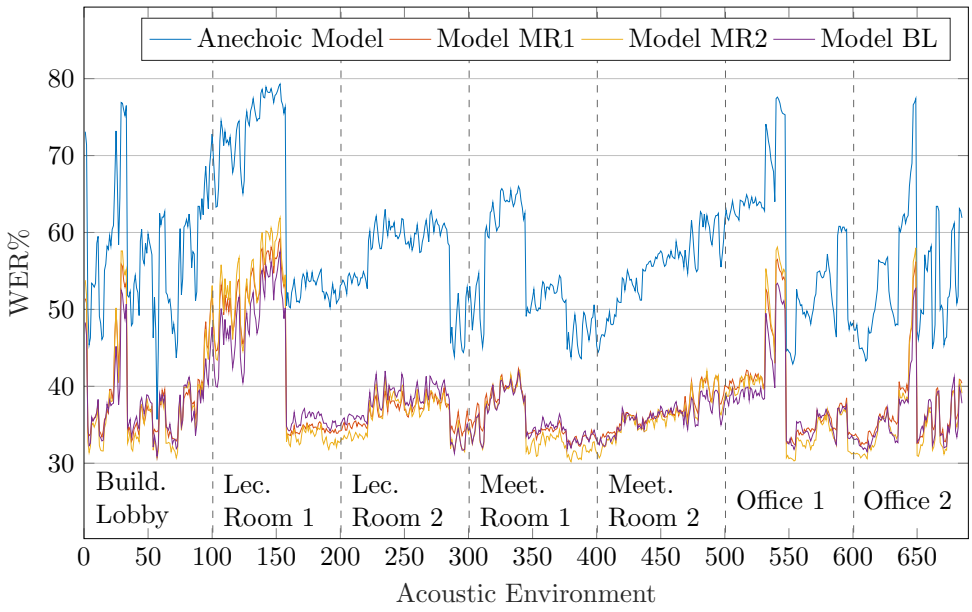


Figure 5.5: WER of ASR decoding of reverberant speech using 4 different acoustic models. Points on each line indicate an AIR and the value at each point is the average WER across all test speech utterances when convolved with the AIR. Each line represents an acoustic model, trained by convolving the train set utterances with AIRs from the room shown in the legend. The acoustic model associated with the lowest WER for an AIR, i.e. the line with the smallest value at the given point, is the label of that AIR for the training of the classifier.

After this step, each one is assigned a label, which is the acoustic model that provides the lowest WER value. The WER values collected for the experiment are provided in Figure 5.5. For the training of acoustic models and running the experiment, the Kaldi Toolkit[2] and the TIMIT[3] speech database are used.

The above Sections have discussed the 4 tasks that are studied in this Chapter. They involve 3 environment classification tasks and the selection of an acoustic model for the decoding of reverberant speech. The experiment setup described in the previous Sections outlines the classifiers, the feature selection method and the candidate features for each task. The evaluation method was presented based on cross-validation, with each fold used to evaluate the accuracy of designed classifiers when using the selected features. The next

[2] `http://kaldi-asr.org`

[3] The s5 example for TIMIT is used as provided with the Kaldi Toolkit at `https://github.com/kaldi-asr/kaldi/blob/master/egs/timit/s5/run.sh`.

Sections review the result of running the relevant experiments. The objective is to identify the subset of features that lead to the most accurate classification.

5.4 Results

Classifying acoustic environments has been addressed in the literature before [33], [117]. A common choice however was to choose a static set of acoustic parameters and evaluate its accuracy on one specific task. The experiments presented in this Chapter take a different approach by considering a variety of candidate features and classifiers, which are evaluated as solutions across a number of tasks. Therefore, novel results are presented in this Section, which are based on the relevant experiments. The results lead to the proposal of feature domains that find multiple applications for the task of classifying acoustic environments.

5.4.1 Baselines

The baseline classification accuracy for each task is evaluated using classifiers that use only 1 of the 4 parameters sets (MFCC, FDRT, FDDRR and AR coefficients) studied in this Chapter. Static feature sets are a representative baseline as they are common practice in the literature for the tasks in question [33], [117]. The results of fitting the classifiers to the data for each parameter set are given in Tables 5.4.

Table 5.4 shows the baselines scores for each task. It shows that a misclassification rate of 0% is given by FDRTs and the kNN classifier for both room and room type classification. The MFCCs also provide near excellent scores, with a misclassification rate of 1.90% for room-type classification and 3.81% for room classification. The corresponding classifiers are the kNN and NBC-GMM respectively. The AR coefficients give the worst scores for both tasks, with misclassification rates that exceed 40%. Trying to identify simultaneously the room and the position in the room is a significantly more challenging task. The misclassification rates for MFCC, FDRTs and FDDRRs are 15%, 15% and 16% respectively. The most challenging task is ASR model selection. Similar scores are given by all parameters except the AR coefficients, which provide the worst score. The lowest

	Feature Domain		NBC Norm.	NBC GMM	CART	kNN	SVM	DNN
Type	Baseline	MFCC	7.62	2.86	20.00	1.90	23.81	1.90
		FDRT	22.86	1.90	3.81	**0.00**	1.90	1.90
		FDDRR	37.14	21.90	13.33	8.57	12.38	21.90
		AR	39.05	32.38	41.90	31.43	44.76	39.05
	Feat. Selection		3.81	**0.00**	1.90	2.86	0.95	0.38
Room	Baseline	MFCC	8.57	3.81	14.29	5.71	36.19	7.62
		FDRT	1.90	1.90	2.86	**0.00**	1.90	3.81
		FDDRR	40.95	24.76	22.86	10.48	11.43	35.24
		AR	51.43	38.10	52.38	34.29	43.81	49.52
	Feat. Selection		**0.00**	0.95	3.81	0.95	0.00	0.00
Position	Baseline	MFCC	18.10	27.62	48.57	15.24	75.24	27.62
		FDRT	21.90	32.38	41.90	**15.24**	32.38	34.29
		FDDRR	21.90	26.67	43.81	16.19	48.57	35.24
		AR	50.48	58.10	72.38	51.43	82.86	64.76
	Feat. Selection		20.95	20.95	34.29	24.76	26.67	**14.95**
ASR	Baseline	MFCC	38.24	37.25	43.14	27.45	44.12	28.43
		FDRT	57.84	58.82	**24.51**	28.43	25.49	34.31
		FDDRR	45.10	41.18	38.24	27.45	35.29	32.35
		AR	53.92	49.02	50.00	46.08	42.16	47.06
	Feat. Selection		32.35	29.41	31.37	28.43	**26.47**	28.43

Table 5.4: Misclassification rate (%) for each classifier and feature domain combination across tasks. Baseline scores are given by classifiers that are trained using static domains. The scores are compared with the result of using feature domains constructed using feature selection (FSS method for all except CART and DNN). The emphasised text indicates best baseline score and best feature selection score per task.

baseline misclassification rate of 24.51% is given by FDRTs using a CART. The above findings verify the results in the literature for room classification, where in [33] FDRTs were used to propose the concept of roomprints that are able to uniquely identify rooms. Also, for ASR model selection, in [35] a CART proved very successful for the task.

Observing Table 5.4 shows that the most accurate baseline results are given by kNN classifiers using MFCC, FDRT and FDDRR. This means that environments that belong to the same class are clustered around a number of central points in the corresponding feature spaces. The opposite is true for the solutions involving AR coefficients. The expectation therefore is for MFCCs, FDRTs and FDDRRs to be employed by classifiers relying on Gaussian distributions, such as NBCs, or by CARTs, which linearly segments the space. AR coefficients can be used by DNNs in combination with other features, which

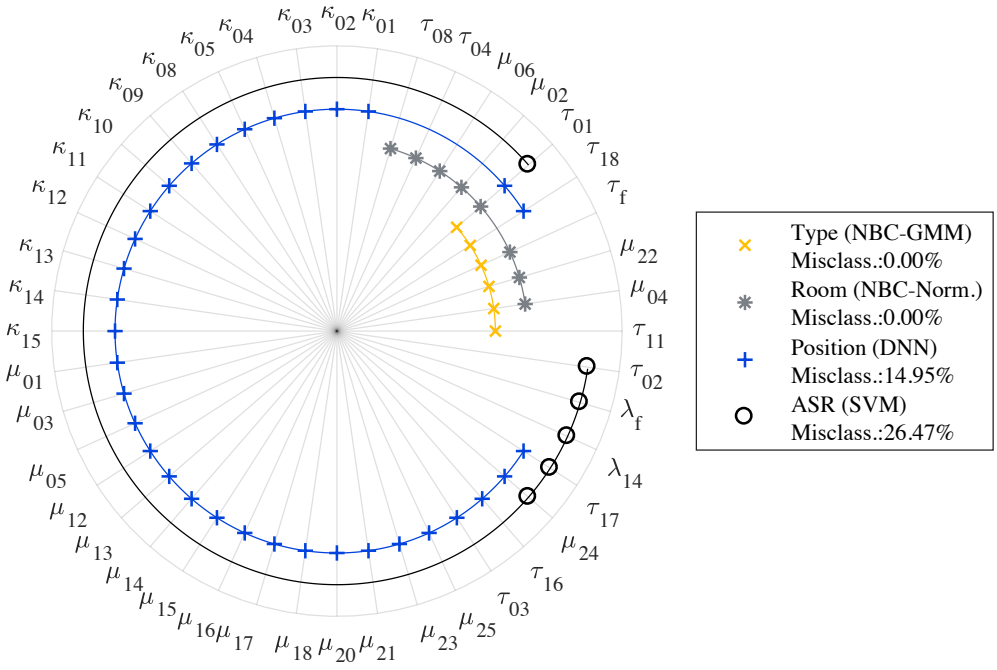


Figure 5.6: Proposed feature domain for each task. Markers along the arcs indicate a selected acoustic parameter. The corresponding classifier types for each domain are given in the legend.

can approximate complicated non-linear functions.

The next Sections use feature selection for the 4 tasks studied. The classification accuracy of classifiers designed using the selected features is compared to the baselines discussed above.

5.4.2 Classification using feature selection

This Section presents novel results that compare the efficacy of using a number of acoustic parameters as features for the classification of acoustic environments. The Section presents how the combination of selected acoustic parameters produces low-dimensional and discriminative feature domains for classification.

Performing the feature selection using the classifiers, acoustic parameters and for each of the tasks considered in this Chapter provides the misclassification rates shown in Table 5.4. The presentation enables an easy comparison with the baseline scores. Figure 5.6

illustrates the features that were selected for each task. The representation uses arcs of concentric circles to represent each task and points on each arc show the features that were selected as part of the domain that gave the lowest misclassification score. This representation creates a compact view of the feature selection results and of the overlap in the resulting domains. Further to the above, for CARTs and DNNs which use embedded feature selection, the Predictor Importance (PI) values are shown in Figures 5.8 and 5.7 respectively. The Figures show the features which collectively contribute 95% of the overall PI in each case. The PI value for CART is evaluated by considering the effect that splits at tree nodes have on the classification risk and associates the risk to the importance of the decision feature at the node [75]. For DNNs it is evaluated heuristically, as the increase in the misclassification rate when feature samples are replaced by WGN with $\sigma^2 = 1$ at the input of the DNN during inference[4].

From the results of Table 5.4 and the feature selection results it can be seen that for room and room type classification, low-dimensional and accurate classification models are given by Gaussian-based classifiers. The Normal distribution and GMM based classifiers give perfect results for room and room type classification respectively. The room classifier utilises only 8 feature dimensions out of the total 78 available. This is a reduction of 56% in dimensionality when compared to using the 18 FDRT coefficients. The room-type classifier utilises only 6, reducing the dimensionality by 67%, again relative to FDRTs. Both domains use only subsets of FDRT bands and MFCCs. The dimensionality of the FDRTs is compared to that of the domains as they also gave a perfect score as a baseline.

For room position identification, the best classification result is given by the DNN. The DNN provides a misclassification rate of 15%. The input to the DNN is formed by all the features, whose weighting is defined by the training algorithm. The PI values in Figure 5.7 show that a mixture of FDRT, MFCC and AR coefficients were important. A closer inspection into the sub-bands used shows that for FDRT the focus is on the bands at the extreme ends of the spectrum. This is attributed to the fact that they contain the least amount of mutual information to each other. Furthermore, in the lowest frequency band,

[4]This approach is proposed in `https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html` and a similar approach is given in [128].

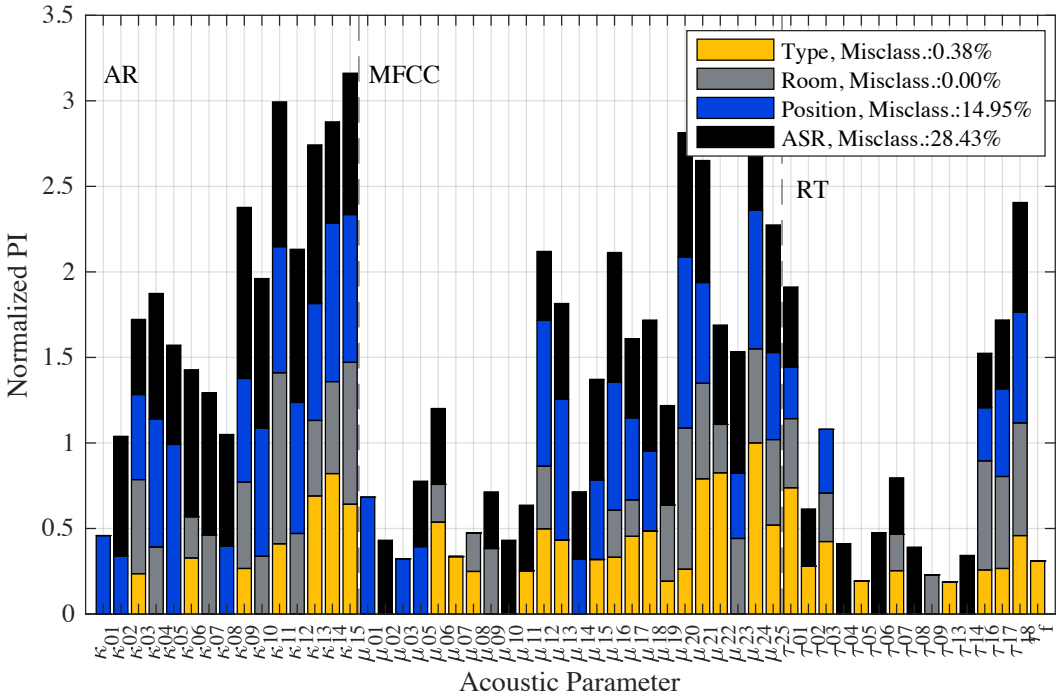


Figure 5.7: Predictor Importance (PI) for DNN for each of the classification tasks considered. It is evaluated heuristically, as the increase in the misclassification rate when feature samples are replaced by Gaussian white noise at the input of the DNN during inference. For clarity, the parameters shown are the ones which collectively constitute 95% of the PI for each task.

room modes are more resolvable [7]. Room modes are related to the shape and dimensions of the room. The RTs at the highest frequency band reveal attributes related to high-frequency absorptions. The AR coefficients carry a significant amount of importance to the predictions made by the network, with almost all the coefficients indicated as useful for the predictions. The network therefore utilises a combination of decay information, spectral features and resonance information to identify which room and where in a room a recording took place.

For the case of selecting ASR models, all classifiers offered similar performance. The best performing classifier given by the feature selection process is the SVM, with a misclassification rate of 26.47%. It uses a combination of FDRTs, MFCCs and FDDRR.

It is interesting to note also that the consideration of combinations of features shifted most of the best scores away from the kNN solutions. More complex relationships between

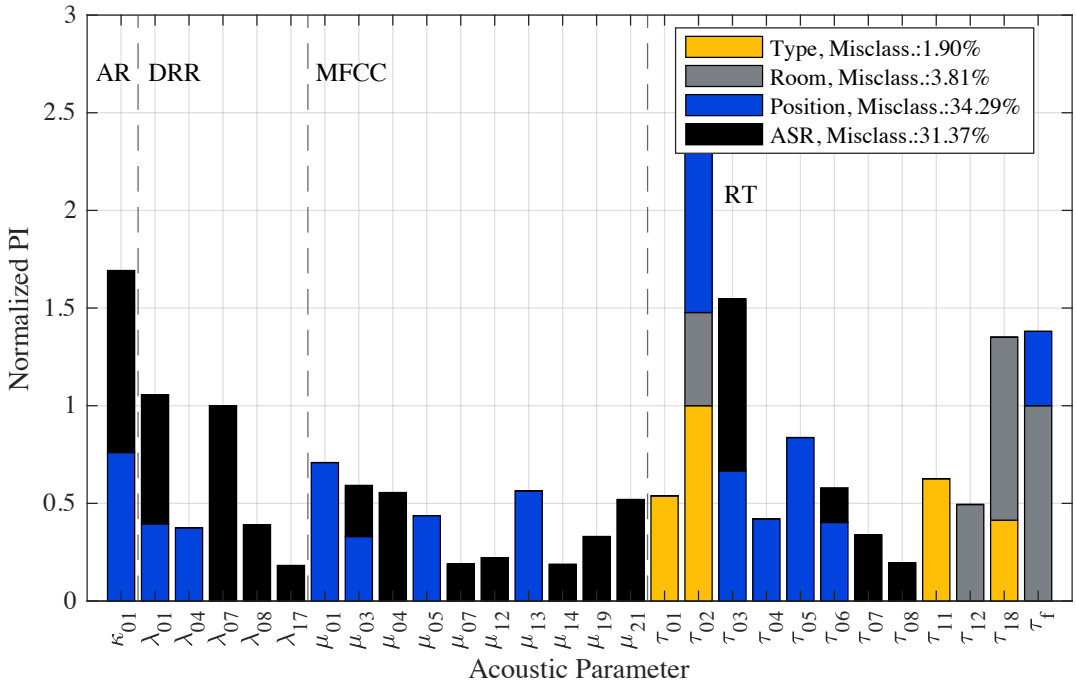


Figure 5.8: Predictor Importance (PI) for CART for each of the classification tasks considered. It is evaluated by considering the effect that splits at tree nodes have on the classification risk and associates them with the decision feature at the node [75].

the feature dimensions have been exploited by other classifiers at this point, which led to accurate and lower-dimensional solutions. The next Section investigates the robustness of these low-dimensional domains to controlled levels of estimation errors.

5.4.3 Robustness of feature domains

Practical applications of the designed classifiers would have as an input a reverberant speech signal. Recalling from Section 2.3, methods exist which allow the extraction of each acoustic parameter from reverberant speech. The estimated features provided by these methods however include estimation errors. A reliable classification scheme will therefore need to be robust to estimation errors, which makes the study on robustness an important part of this investigation.

In order to test the proposed solutions under the presence of estimation errors, errors are artificially added to acoustic parameters. The notation $\mathbf{y}_{k,l}$ is used to denote the

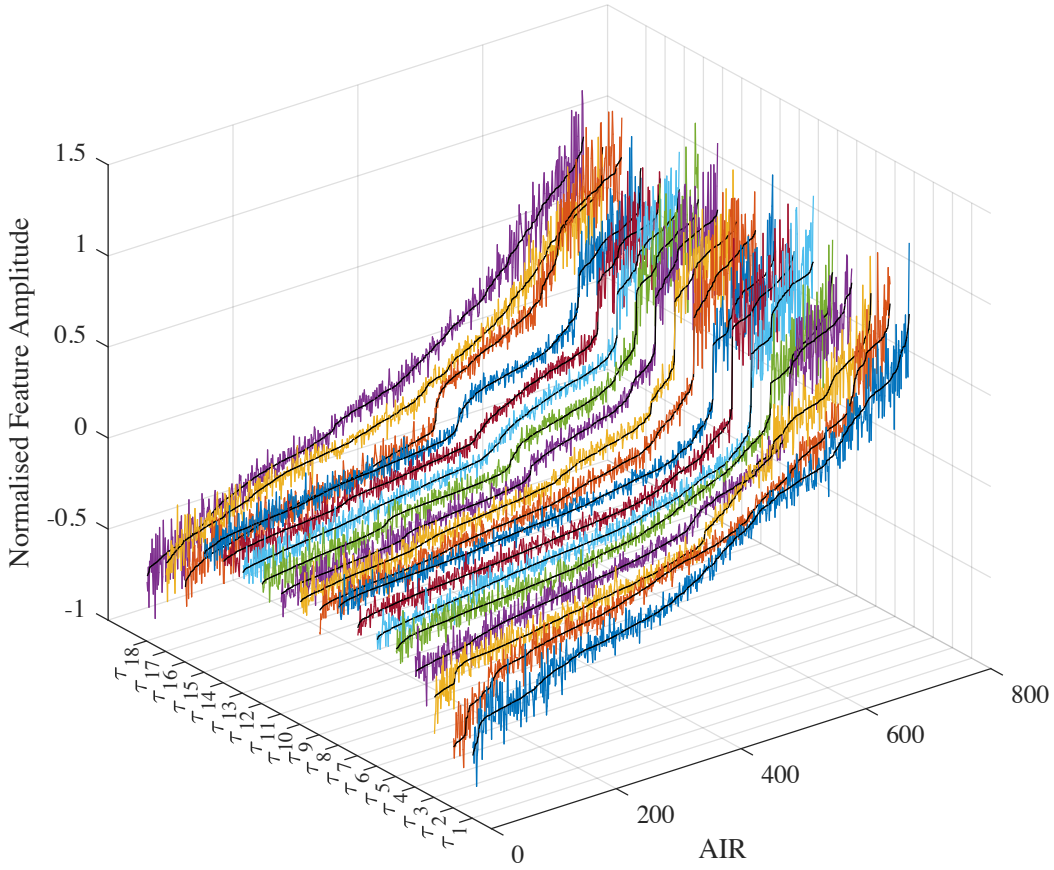


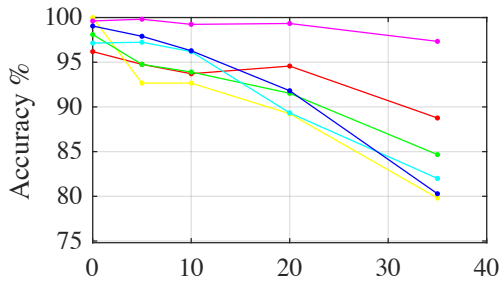
Figure 5.9: Artificial corruption of Frequency-Dependent RT (FDRT) estimates for the purposes of evaluating the robustness of classifiers to additive errors due to parameter estimation. Black lines indicate the ground truth values. The corruption level is set to $\alpha = 20\%$ in (5.11).

column vector containing the values representing the l-th dimension of $\mathbf{Y}_k$ and $\sigma^2_{y_{k,l}}$ to denote the variance of the M elements in that dimension, estimated using

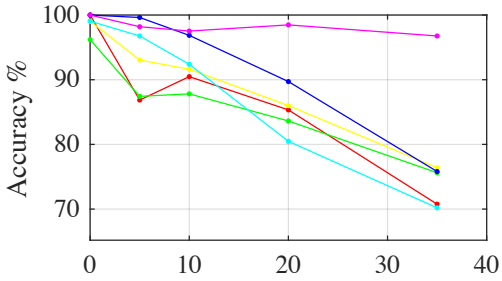
$$\sigma^2_{y_{k,l}} = \frac{1}{M-1} \left(\mathbf{y}_{k,l} - \bar{\mathbf{y}}_{k,l}\right)^T \left(\mathbf{y}_{k,l} - \bar{\mathbf{y}}_{k,l}\right). \tag{5.9}$$

The model of $\mathbf{y}_{k,l}$ that includes an error term added to simulate the presence of estimation errors is

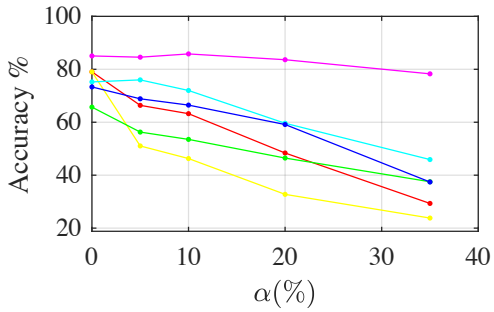
$$\boldsymbol{\epsilon}_{k,l} = \mathbf{y}_{k,l} + \left[e^{(1)}_{k,l}, e^{(2)}_{k,l}, \ldots, e^{(M)}_{k,l}\right]^T, \tag{5.10}$$



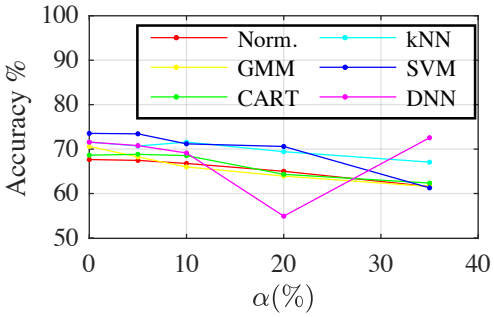
(a) Room-type classification performance at corruption levels.



(b) Room classification performance at corruption levels.



(c) Room position classification performance at corruption levels.



(d) Acoustic model selection performance at corruption levels.

Figure 5.10: Prediction accuracy of classifiers using the proposed feature domains with the test data artificially corrupted at the set corruption level α.

where the random variables $e_{k,l}^{(m)}\ \forall\ m \in \{1,\ldots,M\}$ are defined as $e_{k,l}^{(m)} \sim N(0,\sigma_e)$, where

$$\sigma_e = \begin{cases} \max\left(\alpha\left|y_{k,l}^{(m)} - \bar{\mathbf{y}}_{k,l}\right|, \dfrac{\sigma_{y_{k,l}}^2}{10}\right), & \text{if } a > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

These variables represent the errors due to uncertainties present in the estimation of parameters. The assumption for this model is that the error signal is additive, zero-mean, non-stationary and Gaussian with a standard deviation proportional to the extremity of the true value of the parameter that is being measured. In order to account for other effects and prevent parameter values close to the mean from appearing as error-free, a minimum value has been set for the standard deviation of the error distribution. The scalar α regulates the level of the errors with high values simulating higher estimation errors. Applying this artificial corruption to extracted values of the FDRTs for the AIRs

of the ACE database yields the results of Figure 5.9.

The impact of corruption on the classification performance of the solutions proposed in Section 5.4.2 and Figure 5.6 is shown in Figure 5.10 for each task. The results show that all classifiers except the DNN are severely affected by the impact of noise on the data. Adding noise to the data using (5.10) shifted points in random directions in the feature space, impacting the classification accuracy. If the true distribution of the data *in the wild* is contaminated by noise levels as the ones simulated in this experiment, then these classifiers have overfitted the training data. The DNNs' robustness is attributed to the fact that it accepts a diversity of inputs, unlike the rest of the classifiers that only have as inputs the selected parameters. DNNs are known to overfit, however in this case the training of shallower networks for a small number of epochs with many inputs of different semantic meaning led to a very robust and generalisable solution.

This discussion on the robustness of the proposed feature domains concludes the experiments of this Chapter. The experiments have introduced a set of baselines, based on the choices made in the literature and compared them with a set of designed classifiers. The experiments were repeated for 4 tasks, allowing a direct comparison of the discriminative powers of acoustic parameters across the tasks. The results of the experiments show that, for a set of tasks, the proposed domains offer a lower-dimensionality than the baselines, without reducing the accuracy of the predictions. The next Section provides a discussion on the results and a conclusion.

5.5 Discussion and conclusion

This Chapter proposed feature domains of low-dimensionality, which enable machines to classify reverberant acoustic environments. The proposed domains were proposed through the analysis of novel results given by experiments that compared the discriminative powers of a set of acoustic parameters, across 4 classification tasks, using a number of classifiers.

The experiments have shown that discrimination in terms of rooms and between rooms of different types is effective using sub-bands of the FDRT or MFCCs or a combination

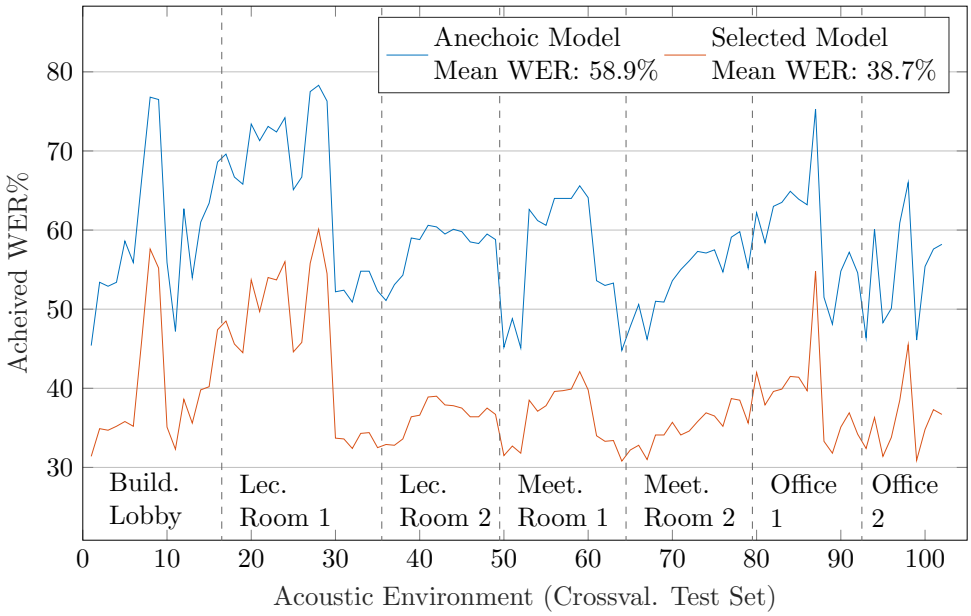


Figure 5.11: Illustrating the difference in the WER for the test environments between the case of using a static anechoic model and the case of selecting a model using the proposed acoustic model classifier.

of the two. Classification in these domains using NBCs, led to a cross-validation accuracy of 100% for the two tasks. It was shown later however that with increasing levels of estimation errors, DNNs offer more robust solutions. This would be at the expense of computational resources, as all the acoustic parameters investigated have to be estimated and passed to the DNN. The choice of the classifier for relevant applications must be made by considering the desired level of robustness given the accuracy of the parameter estimation method used. In terms of room position, the classifier which gave a balanced trade-off between accuracy and robustness in the feature selection experiments is a DNN. The cross-validation accuracy in relevant experiments was 85%. Information from all parameters was used for the classification.

The best acoustic model selection classifier for ASR was given by the SVM, using a combination of MFCCs, FDRTs and FDDRRs, with an accuracy of 74%. The experiments therefore showed that spectral aspects, the energy decay rate and the ratio between the direct and reverberant energy are all important factors in determining which model to use for ASR. In order to demonstrate the advantages of using the proposed feature domain

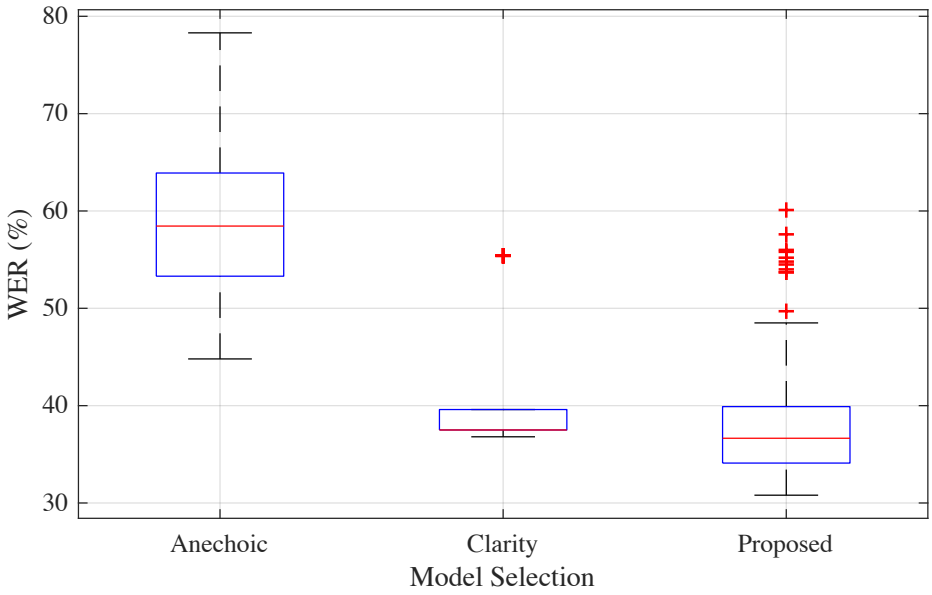


Figure 5.12: Comparing the WER resulting from decoding reverberant speech utterances using an anechoic acoustic model versus selecting from a set of reverberant acoustic models. Two methods are shown for selecting between the models. The first is based on [35] and uses the clarity index C_{50}, whereas the second is an SVM operating in the proposed domain.

for ASR model selection, Figures 5.11 and 5.12 are given. In Figure 5.11, the WERs are compared for the case of performing ASR using a single anechoic acoustic model and for the case of choosing between multiple models using the designed SVM classifier. The results show that using a single anechoic model yields a mean WER of 58.86% and by using the SVM classifier with multiple models the value reduces to 38.69%, a reduction of 20.17 percentage points. For reference, the mean WER in the case of anechoic test and train data is 21.1%, which shows how detrimental reverberation is to ASR. The results also show that the WER is reduced for all the test scenarios, which shows that the anechoic model is never selected. In Figure 5.12, the proposed domain is compared to the case of selecting an acoustic model using the clarity index C_{50} [10], as proposed in [35] for the same task. The model is selected as the one with the closest C_{50} to that of the test condition. The comparison shows that picking a model using the proposed domain leads to the lowest median WER of 36.65%. For the case of selecting a model using the C_{50}, the corresponding value is 37.50%.

An additional important observation is the degree of dissimilarity in the parameters which construct each domain for each task. The significant difference in the domains for room-type identification and ASR acoustic model selection indicates that the usefulness of the descriptor of the acoustic environment is relative to the task. In this distinct example, all but one of the parameters that provided excellent cross-validation accuracy for the former task are disregarded for the latter. The resulting domain still provides an accuracy of 74% for the corresponding task. This highlights the benefit of targeted parameter-extraction as it would not only improve the computational and memory efficiency of relevant applications but would also not compromise performance.

In conclusion, this Chapter presented an analysis of the suitability of a set of acoustic parameters as features, part of discriminative domains, for the classification of reverberant acoustic environments. Environment identification tasks and the task of acoustic model selection for ASR were considered. Using feature selection methods for classification led to the formation of a proposed feature domain for each of the tasks studied. The results of this work provide clear insight for future work in terms of which acoustic parameters are inherently relevant to the discrimination between acoustic environments.

Chapter 6

End-to-End Discriminative Models for the Reverberation Effect

This Chapter proposes a method for the training of generalisable DNN classifiers, able to discriminate between reverberant rooms based on reverberant speech inputs. A DNN architecture is also proposed for the task. The experiments presented provide novel results with regards to the performance of DNNs on the task of room classification and as to the features learned by the networks. This provides insight that is useful in a variety of tasks, beyond room classification.

The task of room classification finds application in forensics [33] and also enables machines to locate themselves amongst known acoustic environments, helping them better understand their surroundings. Chapter 5 showed that handpicking acoustic parameters as features for the task is challenging as their estimation from reverberant speech is known to be problematic [34]. A great benefit of DNNs is their ability to operate in an end-to-end fashion, hence not requiring the extraction of handpicked features [63]. Allowing therefore machines to extract the most useful information directly from reverberant speech inputs can improve the classification accuracy by overcoming the above issues. This Chapter investigates the use of end-to-end room classifiers based on DNNs and offers a comparison of the results with handpicking acoustic parameters as features for the task. To the best of the author's knowledge, there is no existing work in the literature at the time of writing

of this thesis that investigates the use of deep learning for the task of room classification.

Four DNN architectures are considered, using the state-of-the-art in the field, whose performance is compared when trained using the proposed method. The proposed training method presents the data to the network during training in a way that enables the network to learn generalisable properties of the data. The small availability of AIRs, their high-dimensionality and the effect of imbalances of different modalities in the data are obstacles to overcome, both during the design of the network and also during training. The experiments shown involve measured AIRs from real rooms from the ACE challenge database [34] and speech from the TIMIT database. The results of the experiments are used to evaluate the prediction accuracy of the candidate architectures and the most accurate model is proposed for the task.

A further contribution of this work to the field is the insight given by the analysis of the feature-maps extracted by DNNs, both from reverberant speech and by AIRs. These indicate information in the inputs that is identified as discriminative by the networks with regards to different rooms. This information is compared to the handpicked choices of Chapter 5 and links between the two are established. This insight offers many possible avenues to explore in the future, such as how the task of classification and parameter estimation [129] can interact and how transfer learning can be used between the two to improve their performance.

The structure of the remainder of this Chapter is as follows: In Section 6.1 the model for the reverberant speech signal and the training examples used for the training of the models are presented. In Section 6.2 the candidate architectures are discussed and justifications are given with regards to the choices made. The choice for the method for optimising the model's weights is also explained, along with steps that are taken to overcome overfitting. The experiments that evaluate the performance of the DNNs for the task of room classification are presented in Section 6.3. Further discussion and analysis of those results is done in Section 6.4, where a conclusion is also given.

6.1 Signal model and training examples

The motivations and aims of this Chapter are outlined in the previous Section. This Section introduces the notation used throughout this Chapter to denote the inputs and outputs of the classifiers.

Similar to Chapter 5, the reverberant speech signal is defined as $\mathbf{x}$ from (5.2), with samples $x(n)$. The vector $\mathbf{h}$ denotes the AIR between the source and the receiver and $\mathbf{s}$ the anechoic speech signal. The index n refers to the sample index. For the purpose of describing the above for an environment with index m, the corresponding training example will consist of $\mathbf{x}_m$, $\mathbf{h}_m$ and $\mathbf{s}_m$, with the associated class label c_m.

All vectors are transformed into the log-power Fast Fourier Transform (FFT) domain after being segmented into frames. A similar transformation on the input was done for channel classification using DNNs in [6]. The vectors are first split into frames of N_s samples. The framesize is indicated for relevant experiments in later Sections. This operation and the subsequent FFT and power transformation, result to the 3-dimensional arrays $\mathbf{X}_m$, $\mathbf{S}_m$ and $\mathbf{H}_m$, for training example m. The resulting arrays have $\frac{N_s}{2}+1$ elements in each row, which is the size of the one-sided FFT of frames of size N_s. The number of rows N_f, i.e. the number of frames, for each array depends on the number of samples in each of the original vectors.

6.2 Discriminative DNN models

With a notation defined for the environment training examples, this Section introduces the DNNs which are used as classifiers in this Chapter. An introduction to DNNs was given in Section 3.1.2. This Chapter will focus on how the model architectures are created by combining different layer types and how the DNNs can be trained with regards to classifying acoustic environments, using the provided training examples.

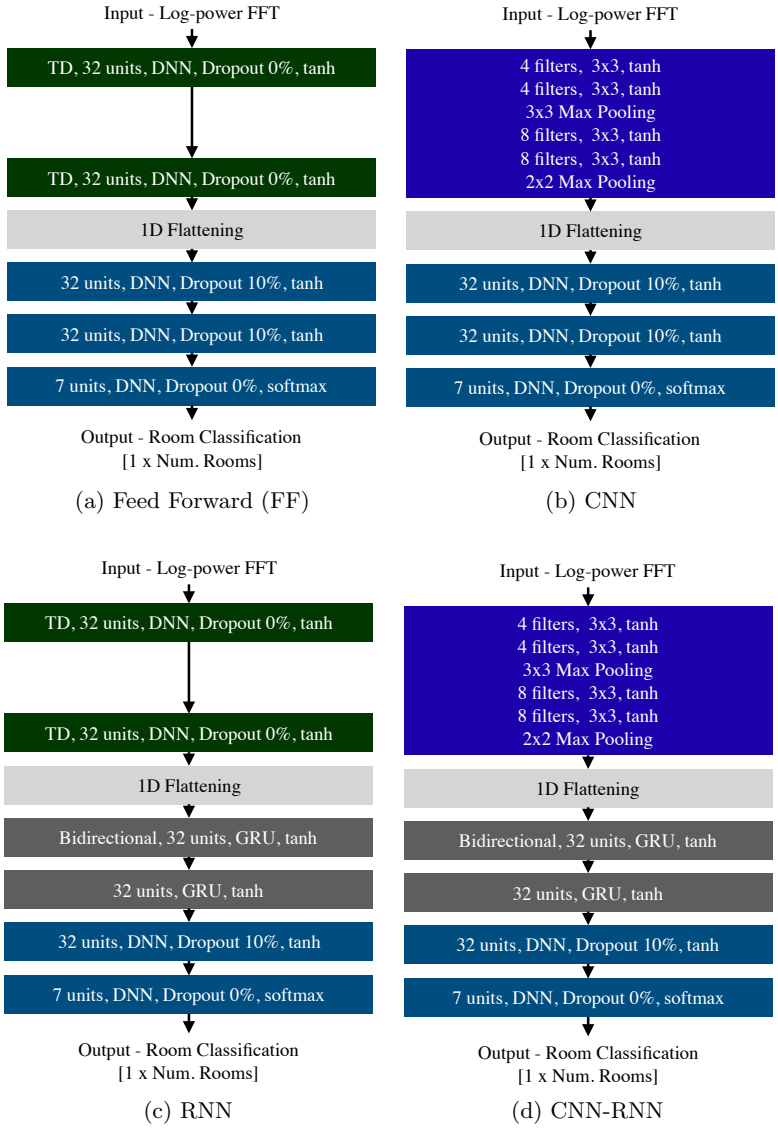


Figure 6.1: Candidate architectures for room classification from AIRs.

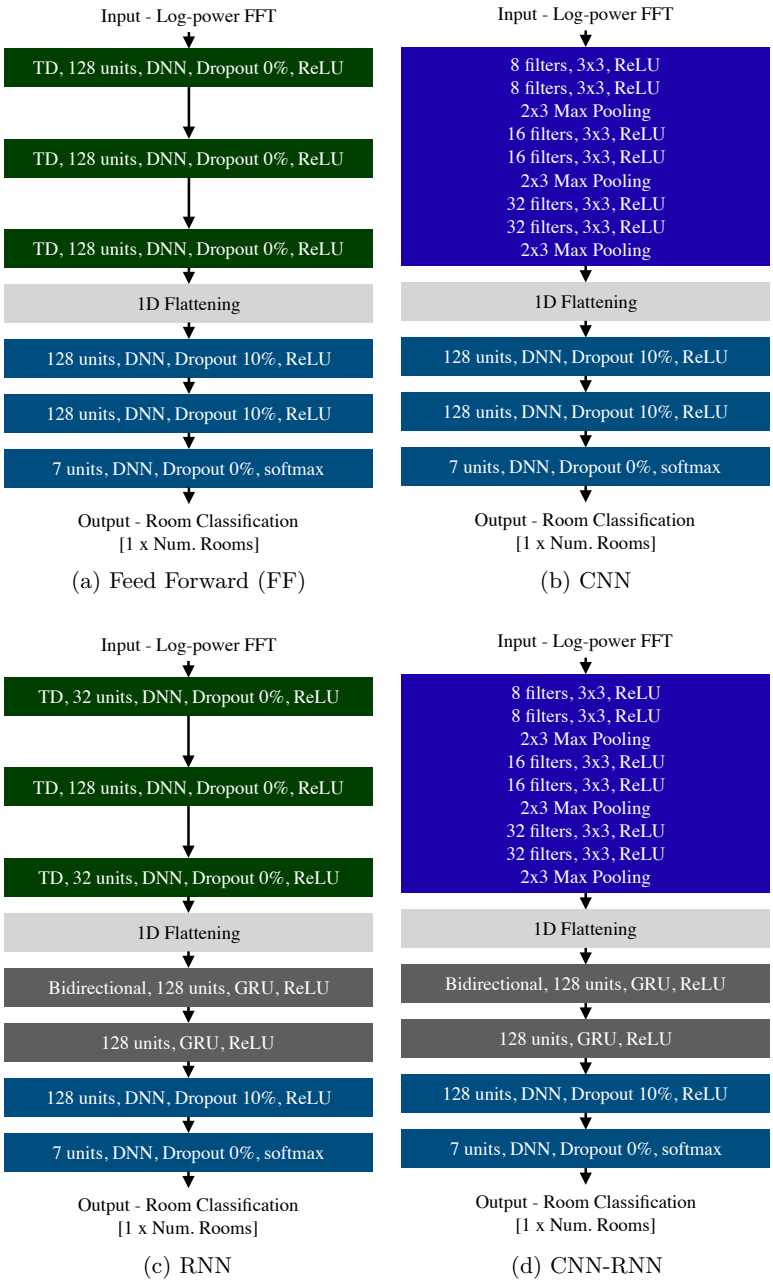


Figure 6.2: Candidate architectures for room classification from reverberant speech.

6.2.1 Candidate DNN architectures

Four candidate architectures are considered, which explore the use of structural choices made in the modern literature. Candidate models are presented in two configurations, one for AIR and one for reverberant speech inputs. The configurations are fine tuned throughout the experiments to highlight the capabilities of each architecture by adjusting the number of layers and the number of neurons in each one. The fine tuning process is rigorous, time-consuming and based on empirical knowledge [36], therefore its documentation is not included.

The four different DNN architectures studied are the following and they are characterised by the types of layers the incorporate.

- *Feed Forward (FF).* This architecture consists solely from FF layers. The first layers process frames in a Time Distributed (TD) fashion. A TD layer processes each frame independently in a memoryless fashion, the same way a FF layer would process each individual frame. The output of the layer has a number of frames equal to its input and the dimensionality of each frame is equal to the number of neurons forming the TD layer. This can be thought off as a frame-based filtering process. Following this, a flattening process converts the frame sequence into a one-dimensional vector, which is processed by subsequent layers. No convolutional or recurrent connections exist in this network.

- *Convolutional Neural Network (CNN).* In this case, the TD layers are replaced with convolutional layers, which filter the input repeatedly providing higher level representations of it. The architecture is inspired from the [62] and [46] models, which have been pioneers in the field of object recognition from images. Their use for the task of Scene Classification [130] has been also very successful in recent work of [131], [132]. The task is related to environment classification.

- *Recurrent Neural Network (RNN).* Looking at the input as a sequence, this architecture replaces the TD layers with recurrent layers with GRU cells. Recurrent layers have been very successful in tasks such as the translation of sentences from

one language to another [49], [65]. Their usefulness in a Bidirectional form [63] for speech recognition was highlighted in [133] when combined with a CTC output layer [48]. The use of bidirectional recurrence is used to create bidirectional GRU layers. These layers create a cell pair, one for each direction of forwards and backwards, which process the sequence in the respective order.

- *CNN-RNN.* Combining convolutional and recurrent layers was studied in [66]. It has been a popular choice across audio and speech processing areas recently. For the case of ASR, this has been done in [134] to jointly train multichannel enhancement and acoustic modelling. For the task of SED it was successful in [107] and for the task of media-presence detection, it was used in [6].

This incremental change in complexity of the models, over the simplest FF architecture, aims to provide insight into its trade-off with accuracy. For the case of AIRs as inputs to the models, the architectures are given in Figure 6.1 and for the case of reverberant speech inputs the corresponding Figure is 6.2.

CNN stages are formed with inspiration from the work of the Visual Geometry Group (VGG) group in Oxford University in [62]. The network is often referred to as the VGG network and involves filters with square kernels of 3×3 dimensions. The filters are layers which are stacked in pairs and pairs are separated by Max Pooling layers [135]. The number of filters between pooling layers doubles as the network becomes deeper. This has been the basis of the convolutional operators in the designs throughout this Chapter. The dropout method [136] is used at the output of FF layers of the networks to improve generalisation. The ReLU activation function [54] is used in this work for larger models as it leads to more effective and efficient implementations by switching off neurons entirely [47]. Its benefits are futher described in [56]. For smaller models, which are proposed here for the classification of AIRs, the tanh activation is used as ReLUs can lead to unstable training, depending on the initialisation. For larger networks with more units it can be assumed that at initialisation approximately 50% of the units will be active [56] however when the network is small and ReLUs are combined with dropout then this assumption is no longer valid, which causes early stopping mechanisms to terminate the

training prematurely. Setting the patience for termination higher can alleviate these issues however it causes longer training times without actually addressing the issue. The tanh function is therefore used for models with AIR inputs. A softmax activation is used for the output neurons, which is given by (3.2).

The four candidate architectures presented above will be evaluated in terms of their accuracy in later Sections. The results of the evaluation will be used to propose an architecture for the task of room classification. The next Section discusses how the DNNs are trained using the proposed method.

6.2.2 Model training

Having defined the model architectures, the next step is to define their training method. The following two aspects are considered in the process:

- Fast convergence of the loss function to a minimum.
- Generalisation of the model to new data.

The discussion below outlines the training method, which considers both aspects.

Parameter optimisation

Recalling from Section 3.1.2, an optimisation algorithm is required to update the weights of the network during training by back-propagating the gradients of the loss function. Pure Stochastic Gradient Descent (SGD) [57] is less commonly used in modern methods as enhanced variants of it have gained popularity. The Adam optimiser is used in this Chapter as proposed in [59]. It offers improvements over AdaGrad [137] and RMSprop[1]. Although all 3 algorithms offer adaptive learning rates for each parameter, which improves training, the review in [138] gives arguments for the benefits of using Adam over its predecessors. In experiments in later Sections, the cross-entropy loss of (3.4) is minimised by Adam as the objective is categorical classification.

[1] An unpublished algorithm described by Geoffrey Hinton as part of an online course`http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`

Model generalisation

Generalisation of the network is crucial and should be ensured through the training process as overfit prevention. Not doing so will likely result in models very capable of describing the training population but of limited practical use. A number of measures are taken as part of the training of the proposed models for this purpose, which are discussed below.

Dropout is incorporated into the training of FF layers of the network. The dropout strategy proposed in [136] turns the output of neurons to 0 with a given probability, which is called the dropout rate, hence preventing the co-adaptation of feature detectors. The dropout rate for FF layers is indicated for each in the Figures 6.1 and 6.2. Early stopping is also used to terminate the training when either the training or the validation loss have not improved for 50 epochs. The validation set is created by using a random stratified selection across recordings and it is used to prevent overfitting. The final network weights are captured at the epoch where the validation loss is at its minimum.

As with many other sound classification tasks [139], the task of environment classification involves training using imbalanced data sets. This imbalance can be in terms of classes. For example in a room classification task, it is likely to have 10 example recordings from Room A and 1000 example recordings from Room B. Naturally, the classifier trained using this data would find it more rewarding to overestimate the chance of observing Room B over Room A. This can be prevented if this imbalance is considered during training. The nature of the imbalance can also be in terms of modalities in the data other than the class labels. If for instance multichannel recordings are to be classified on a per-channel basis, again for a room classification task, then a receiver which has 32-channels would have a higher test score than a receiver with 1-channel, simply because the model was rewarded more for learning properties of the receiver that offered more training data. This imbalance not only gives more data for specific receivers but also for specific receiver positions in rooms, where the 32-channel receiver was placed. Several other sources of imbalance can arise, which are side-effects of the way that data was measured. Other sources of imbalance are related to the characteristics and position of the source, the configuration of a room and noise sources.

This Chapter aims at compensating for the effect of imbalances in the training examples for modalities relating to receivers and receiver positions during training. This will allow for the models' performance to be invariant across different receiver and receiver positions and generalise to new data. The imbalances are accounted for by ensuring that each batch passed for training involves an equal number of AIRs from each receiver, each position and each room. Therefore, each group of each of the above has an equal contribution to the gradient used for each weight update.

This Section defined the candidate DNN architectures for the task of room classification and the proposed method for their training. The next Section presents the process of collecting and preparing training and test data. Also, experiments are carried out to evaluate the performance of the resulting DNNs for the task of room classification.

6.3 Experiments

The experiments described in this Section provide novel results with regards to the performance of DNNs on the task of end-to-end room classification. This Section first presents the accuracy of classifying AIRs in terms of the room in which they were measured in, using the candidate DNN architectures. The experiment is then repeated using reverberant speech. This time, the experiment does not assume that the classifiers have access to any AIR data during training or inference. The models are trained using the method proposed in this Chapter and the most accurate architecture amongst the candidates is proposed for the task.

Chapter 5 looked at handpicked features based on acoustic parameters as a solution to the same problem. The evaluation of the classifiers studied in this Chapter is carried out with the cross-validation procedure and data unchanged for Chapter 5. This allows for a direct comparison of the results. The cross-validation process was discussed in Chapter 5.3.1.

6.3.1 AIRs and speech training-data

The data used for this work are AIRs, which describe acoustic environments. As a source of AIR measurements, the ACE challenge data is used. It contains 700 AIRs, evenly distributed amongst 7 rooms and measured using 5 receiver arrays. 100 AIRs are available for each room, giving a total of 700 AIRs. The distribution of the available data in the database with regards to rooms and receivers is shown in Table 5.2. The speech data convolved with the AIRs to produce reverberant speech is from the TIMIT database. All data for all experiments are resampled to a sampling rate of 16 kHz, the original sampling rate of TIMIT.

With databases for the AIRs and the anechoic speech selected, the training data examples are constructed as follows, using the notation for Section 6.1. From the ACE database, the matrix of AIRs, $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_M]$, is constructed with dimensions $\left[M, N_f, \frac{N_s}{2} + 1\right] = [700, 50, 65]$. The framesize is set to 8 ms (128 samples) for the case of analysing AIRs. All AIRs are zero-padded or truncated to a duration of 400 ms (6400 samples). For each case, the vector of ground truth labels for the data $\mathbf{c}$ is created by associating the AIR with the room in which it was measured in. Reverberant speech is created by pairing each AIR in the training data with a speaker from TIMIT. Anechoic speech utterances of 5 s are created by concatenating shorter utterances from a single speaker. The resulting utterance is then convolved with one AIR. Each AIR from the ACE database is convolved with 20 such speech utterances. Each time a speech utterance is selected for training, it is offset by a random number of samples. The selection for which utterance to convolve with an AIR is random with replacement, which means that the speech data are oversampled. This is considered as a form of primitive data-augmentation. The resulting reverberant speech samples are used to create array $\mathbf{X}$ of dimensions $\left[20M, N_f, \frac{N_s}{2} + 1\right] = [1400, 500, 161]$. The framesize N_s for the analysis of speech is set to 20 ms (320 samples), which gives a trade-off between temporal and spectral resolution. Both are important for the models as the aim is to evaluate the importance of models relying on temporal and spectral aspects of the input.

6.3.2 Training data batches

The method proposed in this Chapter for the training of generalisable DNNs accounts for imbalances in the training data, as discussed in Section 6.2.2. The method forms batches that are passed to the DNNs for training, using an equal number of AIRs from each room-receiver combination. This way, each receiver, each room and each measurement position contribute equally to each gradient calculation. The distribution of the data across rooms and receivers for the ACE database is shown in Table 5.2. What can be seen from the Table is that the data is evenly distributed across rooms and the only imbalance is the number of channels available for each measurement position. The batch construction method therefore needs to balance the batches with respect to the measurement positions. The method includes in each batch a fixed and equal number of channels from each of the 70 total number of measurement positions. This conveniently creates a balance across the positions and simultaneously across rooms, as the 70 measurement positions are split evenly amongst the rooms.

The batch construction starts by shuffling the channels in each of the 70 measurements and putting them in a list. Channels are collected from each one of the lists to construct each training batch. The selection process starts from the top of the list and moves towards the bottom as channels are selected for training. When the list end has been reached then the list is reshuffled and the process is repeated. The shuffling step has been suggested in [140] to improve the training performance and convergence, as well as memory access efficiency. For a variable number of channels per measurement, such as the ACE database, the process is more complex. This is because each list will have a different length. In this case, each list needs to be individually managed and shuffled asynchronously from the rest. For experiments classifying AIRs, one channel from each list is used per batch and for end-to-end classification from reverberant speech, 2 channels from each list are convolved with anechoic speech per batch. The respective batch sizes are therefore 70 and 140.

With the method for collecting the training data and passing them for training defined, the next Sections look at how the trained models are evaluated.

6.3.3 Model evaluation method

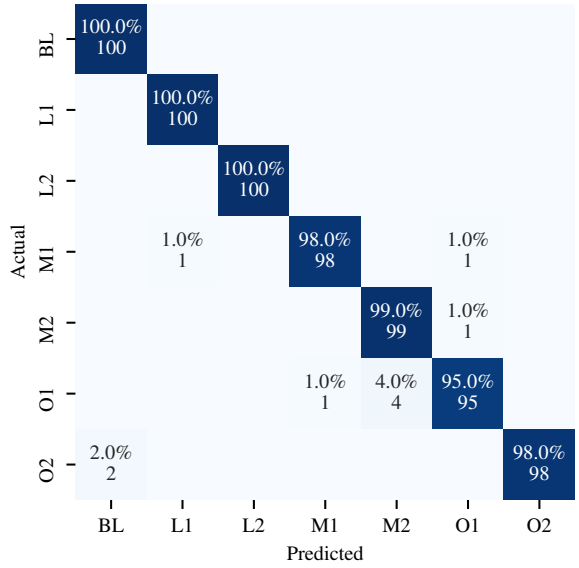
The method used to evaluate the models has remained unchanged from the task of room classification in Chapter 5. Cross-validation is used, with each fold describing one AIR measurement position of Table 5.2. This simulates the scenario of having trained a model given the available AIRs and having to classify a new measurement at an unknown receiver position in one of the known rooms. Each fold is tested by training a model using the out-of-fold samples and predicting the class of the in-fold samples. The validation set used for early stopping is constructed at the beginning of the training of each model using 15% of out-of-fold samples. The selection is stratified and random, as discussed in Section 6.2.2. As the evaluation measure, the accuracy of predictions is used. To summarise the above, with 70 AIR measurement positions in total forming the ACE database, 70 models are trained. Each model is used to predict the room where the AIRs of a left-out measurement position were recorded in. The accuracy of these predictions is eventually used to evaluate each of the proposed models. The process is repeated independently for each of the models of Figures 6.1 and 6.2, for the case of AIR and reverberant speech inputs respectively.

For the case of reverberant speech, each fold involves 20 times more training and test data compared to the case of using AIRs, as each AIR is convolved with 20 speech utterances. This results in approximately a total of 80 hours of reverberant speech[2], evenly split between training and testing. Although the same anechoic speech utterances are allowed to be used multiple times to create reverberant speech, the same speakers are never used for both training and testing. The training and test split of the database is maintained as with the official TIMIT distribution.

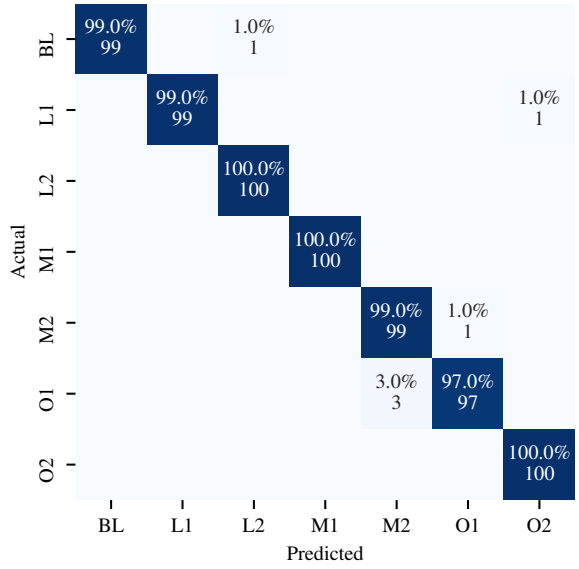
6.3.4 Room classification from AIRs

The result of evaluating the proposed DNNs of Figure 6.1 using AIRs as the inputs is shown in Table 6.1. The table gives the accuracy and the number of trainable parameters for each of the proposed models. The figures in the Table show that the worst performing model in the experiment is the FF network, with an accuracy of 93.3%. This model

[2]Number of AIRs $\times$ utterances per AIR $\times$ utterance length $= 700 \times 20 \times 5$ s ≈ 40 hours



(a) CNN



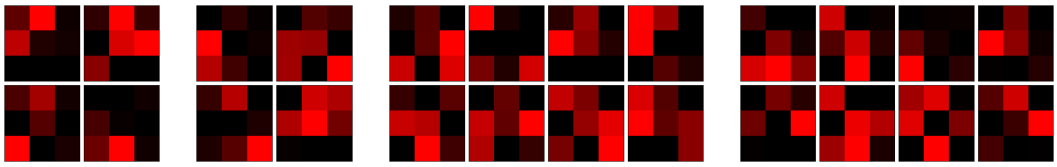
(b) CNN-RNN

Legend: MR1 - Meeting Room 1, MR2 - Meeting Room 2, BL - Building Lobby, O1 - Office 1, O2 - Office 2, LR1 - Lecture Room 1, LR2 - Lecture Room 2.

Figure 6.3: Confusion matrix for room classification based on AIRs from the ACE database.

Model	FF	CNN	RNN	CNN-RNN
Trainable Param.	107,943	44,403	26,247	33,363
Accuracy	93.3%	98.6%	97.9%	99.1%

Table 6.1: Room classification from AIRs, accuracy and number of trainable parameters for each candidate model architecture.



(a) Filter kernels for first convolutional layer.

(b) Filter kernels for second convolutional layer.

(c) Filter kernels for third convolutional layer.

(d) Filter kernels for fourth convolutional layer.

Figure 6.4: Filter kernels for convolutional layers of the CNN-RNN model for room classification of AIRs. Solid black indicates the lowest value.

involves no convolutional and recurrent layers and involves the largest number of trainable parameters. The addition of the recurrent layers increased the accuracy to 97.9% and used almost a quarter of the training parameters. The CNN model gave a prediction accuracy of 98.6% and the combination of convolutional and recurrent layers gave the most successful model, the CNN-RNN with an accuracy of 99.1%. Visualising the errors in Figure 6.3 for the best performing CNN-RNN and runner-up CNN models, shows that the CNN-RNN misclassified only six AIRs from the total of 700. To compare the result with existing approaches in the field, in Table 6.2 the result is compared to the classification methods proposed in [33] and [117]. The accuracy that the CNN-RNN offers is higher than that of both methods. However, the domain proposed in Chapter 5 offered an accuracy of 100% for the same task. Therefore, this experiment has shown that the CNN-RNN can predict which room an AIR was recorded in, with the difference in accuracy being less than 1% when compared to making the same classification by having perfect knowledge of a subset of acoustic parameters.

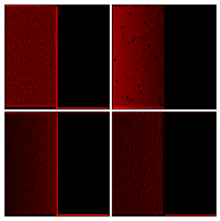
A benefit of convolutional layers is that they allow for interpretable visual representations of the learned feature-maps [63]. This step enables the comparison of what was found to be important in Chapter 5 for discriminating between rooms and what the learned

Classifier	NBC FDRT	GMM MFCC	NBC FS	CNN RNN
Accuracy	98.1%	96.2%	100%	99.1%

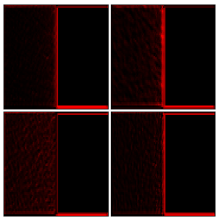
Table 6.2: CNN-RNN room classification accuracy from AIRs compared between a Normal distribution NBC FDRT classifier (as proposed in [33]), a ML-GMM MFCC classifier (as proposed in [117]) and a Normal distribution NBC classifier operating in the domain proposed in Chapter 5. The accuracies for the comparison are extracted from Table 5.4.

feature-maps by the CNN-RNN are. Since the inputs to the network are spectrograms, the network learns a set of filters in the log-spectral domain, which are shown in Figure 6.4. The 3×3 filters operate successively on the spectrogram input to provide higher level features, which are then passed to the Max Pooling, recurrent and FF layers for classification. Although the filters provide no visual cue for their importance, their effect on filtering AIRs is easy to interpret. In Figure 6.5, the result of passing an AIR measured in an office through the CNN-RNN is shown. The Figure shows a contrast between the filter activations around points where the AIR energy has significantly decayed and the activations before that point. Repeating the process for the remaining test AIRs gives similar behaviour. This indicates that the high level features provided by the convolutional layers relate to the level of decay of the sound energy at different frequencies, much like the information provided by the FDRTs. This is in line with the observations from Chapter 5, which indicated that the FDRTs are useful in discriminating between different rooms. This provides a link between the handpicked features and the machine-derived feature-maps.

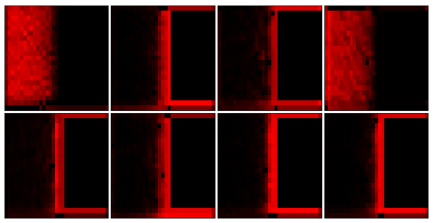
A discussion on the number of parameters involved in each model provides further insight with regards to the performance of the candidate architectures. The simple FF DNN involved the highest number of parameters. It did not use any pooling or downsampling mechanisms like the rest of the models. The number of parameters is an important factor in the performance of the model, as the available data for training in the given experiments is relatively small. Whilst famous models involve millions of parameters, such as the Alexnet [46] that involved 60 million, their training examples are of the same order (1.4 million images available [141] for Alexnet). Larger models are more powerful but



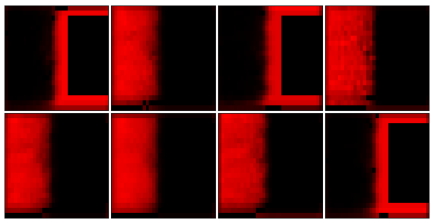
(a) Filtering through the first convolutional layer.



(b) Filtering through the second convolutional layer.



(c) Filtering through the third convolutional layer.



(d) Filtering through the fourth convolutional layer.

Figure 6.5: Filtering an AIR measured in an office room through CNN and Max Pooling layers of the CNN-RNN model for room classification of AIRs. The vertical pixel progression refers to frequency bins and the horizontal pixel progression refers to the frame index, with frequency 0 and time 0 at the bottom left corner. Solid black indicates the lowest value.

require a proportional number of training examples. This is an issue for AIR classification as large databases, of the order of millions, are not available for real rooms. Also, AIRs are high dimensional, which means that a large number of parameters is needed to process them. Convolutional layers deal with the high-dimensionality by processing the AIR as an image, using 2-dimensional filters to transforming it into high level features through successive pooling. This enables the network to detect patterns in the input at different parts of it, using the same trainable parameters, the filer kernels. This property of convolutional layers is referred to as *parameter sharing* in [63]. Similarly, recurrent layers reduce AIRs into vectors by processing their frames as sequences. This link between the nature of the problem and the CNN-RNN model is what its success and its near perfect accuracy is attributed to.

The next Section uses the insight gained from the above analysis and repeats the experiment, this time using only reverberant speech samples. The candidate architectures

Model	FF	CNN	RNN	CNN-RNN
Trainable Param.	8,280,459	1,322,499	417,035	405,891
Accuracy	59.4%	79.1%	83.3%	86.9%

Table 6.3: Room classification from reverberant speech, accuracy and number of trainable parameters for each candidate model architecture.

are again evaluated in terms of their accuracy for the task.

6.3.5 Room classification from reverberant speech

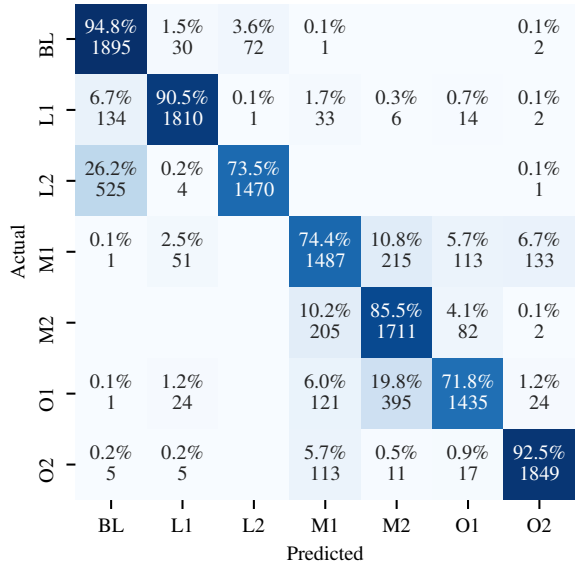
The previous experiment has investigated the accuracy of classifying AIRs in terms of the room in which they were measured in. The candidate architectures for the task were benchmarked in terms of their accuracy in doing so. This Section repeats the experiment but this time using only reverberant speech signals for the classification. The DNNs are not presented with the original AIRs during training or inference.

The results of training DNNs for end-to-end room classification from reverberant speech are presented in Table 6.3. The Table shows the number of trainable parameters for each candidate architecture and their corresponding accuracy for the task. The results show that the FF network has 77 times more trainable parameters than the equivalent AIR-input FF network. This increase is due to the much higher dimensionality of the input, having moved from 400 ms AIRs to 5 s speech signals. The accuracy of the FF network is 59.4%. Adding convolutional layers to form the CNN increases the accuracy by 20%, and uses approximately $^1/_6$ of the parameters of the FF network. Using recurrent layers to form the RNN instead, increases the accuracy of the solution by 23.6% and also uses approximately $^1/_{20}$ of the parameters of the FF network. The best performing network is the CNN-RNN with an accuracy of 86.9%. It involves approximately $^1/_{20}$ of the parameter count of the FF network. The CNN-RNN model for AIR classification was also the most successful in the previous experiment, with an accuracy of 99.1%. The CNN-RNN with reverberant speech input uses 13 times more parameters than the CNN-RNN using AIRs. This shows the increase in complexity related to training networks able to process reverberant speech directly. The confusion matrix of testing using the two best

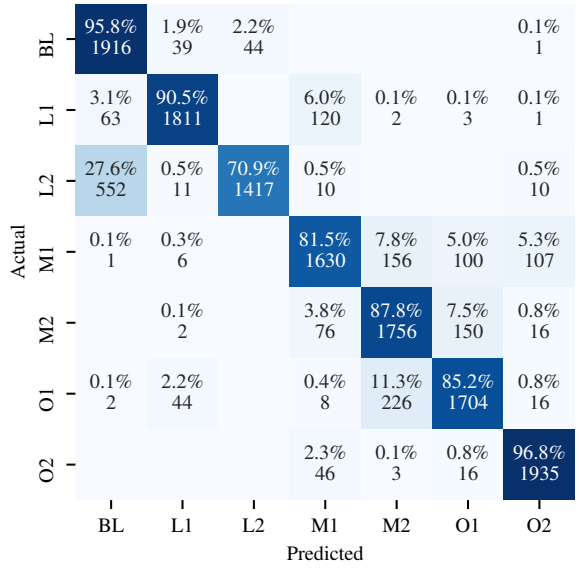
performing models, the RNN and CNN-RNN models are given in Figure 6.6.

The number of parameters of the resulting networks is significantly greater to the AIR case, across all architectures. This scaling of the DNNs in terms of trainable parameters and the reduction in the highest accuracy by 12.3% shows the increased challenges in performing the classification directly from reverberant speech. As introduced in Section 2.1, the AIR describes an acoustic environment. Therefore, classifying AIRs directly is a task where a representation of the environment is to be transformed to a lower-dimensional space, similar to [114], [116]. Performing the classification from reverberant speech is significantly different however as the objective remains the same but the input to the network is the result of the interaction of the environment with an unknown speech signal. Since the spoken utterances and speakers are not correlated with the rooms' properties, any model inevitably needs to decouple the speech information from the channel information to make any meaningful predictions. The general conclusion is that the increase in the number of parameters and the reduction in accuracy is attributed to the additional task of separating speech from channel information. Using speech instead of AIRs encodes information about the environment in the reverberant speech signal but can also cause some of the information to be lost. As speech is not broad-band in windows of a few milliseconds, longer frame sequences are needed to provide significant information to the system across the spectral range of interest. The duration of the input to the network increased from the order of milliseconds to seconds. Even in these longer sequences, and in fact in the sequence of any length, there is no guarantee that the speech signal will contain substantial energy at all frequencies across the spectrum, which impacts the classification accuracy.

A measure of the success of the decoupling between the speech and channel information is given by examining the accuracy of the best performing model, the CNN-RNN, across a range of speaker characteristics. Tables 6.4 and 6.5 show the variation in classification accuracy per speaker dialect and gender across rooms. The maximum deviation of the accuracy across dialects is 1.35% and 0.05% for gender, relative to the mean accuracy. All dialects and both genders show very similar scores, which illustrates that speaker



(a) RNN



(b) CNN-RNN

Legend: MR1 - Meeting Room 1, MR2 - Meeting Room 2, BL - Building Lobby, O1 - Office 1, O2 - Office 2, LR1 - Lecture Room 1, LR2 - Lecture Room 2.

Figure 6.6: Confusion matrix for room classification based on reverberant speech using data from the ACE and TIMIT databases.

Dialect	Utterances	MR2	MR1	BL	O2	O1	LR2	LR1	All
DR1	906	86.8	81.3	93.1	94.4	74.6	71.8	90.2	84.3
DR2	2228	89.9	81.4	94.1	95.8	81.0	72.3	89.0	86.3
DR3	2184	90.3	75.4	93.3	94.1	81.4	66.7	86.4	84.0
DR4	2691	87.6	79.7	92.1	93.5	79.9	69.3	88.7	84.2
DR5	2401	91.8	80.9	91.8	94.0	82.6	64.2	89.5	85.1
DR6	886	94.6	82.2	95.4	96.8	82.4	65.4	83.0	86.2
DR7	1850	89.8	81.1	91.7	97.5	82.5	67.8	87.4	85.5
DR8	854	90.9	79.1	94.6	95.0	81.2	65.6	87.2	84.7
Mean	1750	90.2	80.2	93.3	95.2	80.1	67.9	87.7	85.0
Overall	14000	80.7	87.0	69.1	92.5	86.6	97.8	94.9	86.9

Table 6.4: Room classification test accuracy (%) from reverberant speech, with regards to spoken dialect.

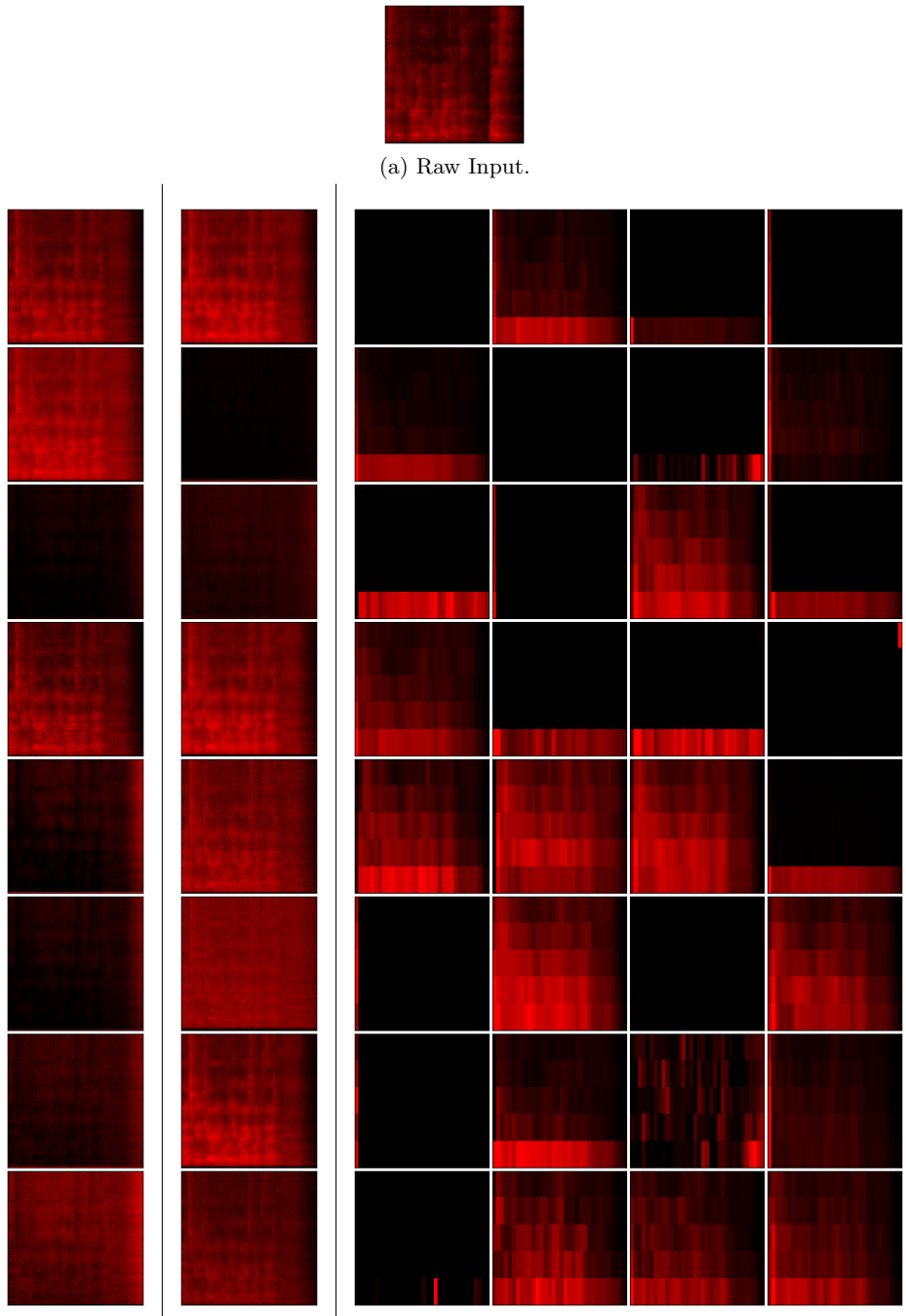
Gender	Utterances	MR2	MR1	BL	O2	O1	LR2	LR1	All
Female	4676	91.1	80.6	93.2	93.9	81.9	67.8	88.7	85.5
Male	9324	89.5	79.8	92.8	95.5	80.5	68.1	87.6	84.7
Mean	7000	90.3	80.2	93.1	94.7	81.3	68.0	88.2	85.2
Overall	14000	80.7	87.0	69.1	92.5	86.6	97.8	94.9	86.9

Table 6.5: Room classification test accuracy (%) from reverberant speech, with regards to speaker gender.

Legend: MR1 - Meeting Room 1, MR2 - Meeting Room 2, BL - Building Lobby, O1 - Office 1, O2 - Office 2, LR1 - Lecture Room 1, LR2 - Lecture Room 2.

characteristics do not significantly impact the result. Observing both Tables 6.4 and 6.5 it can be seen that these variations are significantly different when marginalising along the rooms, which is the true axis of variation to be exploited in this task. The maximum deviation is 20.5%, relative to the mean accuracy.

The CNN-RNN is shown to perform the desired task at a high level of accuracy and also to be substantially invariant to speaker variations. To understand the information that the network uses to do so, the feature-maps extracted in its layers are visualised. Figure 6.5 shows the raw input log-power spectrogram, the corresponding output resulting from filtering through the first 2 convolutional layers of the network and the output of the last Max Pooling layer. Whilst the output of the first two convolutional layers does not appear meaningful to a human, meaningful interpretations of the activations become apparent as the data propagates through the network. What can be seen is that the



(a) Raw Input.

(b) Filtering through first convolutional layer.

(c) Filtering through second convolutional layer.

(d) Filtering through last Max Pooling layer.

Figure 6.7: Filtering a reverberant male speech utterance, created using an AIR measured in an office room, through the CNN-RNN model for room classification of reverberant speech. The spoken utterance has a duration of 2.8 s and the frame length is 50 ms. The transcription is "Don't ask me to carry an oily rag like that". The vertical pixel progression refers to frequency bins and the horizontal pixel progression refers to the frame index with frequency 0 and time 0 at the bottom left corner of each plot. Solid black indicates the lowest value.

resulting activations appear as trajectories in time. Therefore, for most of the 32 output filters, certain regions of the activations, corresponding to frequency bands, are consistently higher than others across frames[3]. This indicates that the frequency dependence of the energy content in the input is exploited by the network. In order to perform the task of room classification from reverberant speech, the feature-maps learned by the CNN-RNN architecture therefore relate to time-invariant properties of spectral regions of the input. This is consistent with both the observations of the previous Section, where the same task was carried out directly from AIRs, and of Chapter 5 where FDRT were used for the classification. This establishes a link between the importance of frequency-dependent time-invariant features for the cases of classifying from AIRs and from reverberant speech.

This analysis concludes the experiments of this Chapter. The experiments have evaluated the performance of 4 candidate architectures for the task of room classification using either AIRs or directly from reverberant speech. The DNNs were trained using the method proposed in this Chapter and the experiments provided novel results with regards to the effectiveness of using DNNs for room classification. The next Section provides a discussion on the findings of this Chapter and offers a conclusion.

6.4 Discussion and conclusion

This Chapter proposed a method for the training of generalisable DNN classifiers, able to discriminate between reverberant rooms, based on reverberant speech inputs or AIRs. The training method has shown how to overcome imbalances in the distribution of channel properties in the data for effective and efficient training. This investigation provided novel results with regards to the performance of DNNs on the task of room classification and as to the features learned by the networks. The performance of the trained classifiers was compared to that of classifiers constructed using combinations of handpicked features. Through the analysis of the results, the CNN-RNN architecture has proved to be the most accurate end-to-end room classifier and it is therefore proposed for the task.

[3]Any time dependence or sparsity in the activations can be attributed to the spectro-temporal variations in the distribution of the energy of the speech signal.

In the experiments carried out in this Chapter, the CNN-RNN trained using the proposed method achieved an accuracy of 99.1%, when training and inference is done using AIRs. The respective handpicked-feature room classifier from Chapter 5 achieved an accuracy of 100% and was based on a NBC, utilising RT information and 2 MFCCs. Performing the same classification task from reverberant speech and assuming no access to the AIRs, the DNN classifier gave a classification accuracy of 86.9%. The corresponding classifier of Chapter 5 has shown in Figure 5.10 to be susceptible to estimation errors of its input parameters. The parameter estimation from reverberant speech is expected to involve estimation errors [34], inevitably impacting the accuracy of the predictions. Performing the same task from reverberant speech using an end-to-end DNN does not suffer from such issues, since there is no need to estimate any acoustic parameters and no domain expertise is needed. Feature-maps are derived by the network directly from the reverberant speech signal and used by subsequent layers for the task of classification.

A link between the feature-maps extracted by the DNNs and the handpicked features, namely the FDRTs, was established in the experiments. Given this observation, a direction for future work would be to improve the training of DNN room classifiers and DNN acoustic-parameter estimators [129] by sharing layers between the two types of networks. This can be done at a pre-training stage to speed-up convergence for larger scale experiments and potentially allow for better results.

In conclusion, this Chapter has proposed a method for training DNNs for the task of room classification from either AIRs or reverberant speech. To the best of the author's knowledge, this work is the first step in using DNNs for the task of room classification from reverberant speech. Novel results are given with regards to the accuracy of predictions made by models of different architectures. Based on these results a CNN-RNN is proposed for the task. Insight is given as to what information from the input is identified as discriminative by the network, by analysing the feature-maps at intermediate layers. The experiments in this Chapter involve reverberation data from the ACE challenge database and have shown that with access to the AIRs of the environments, the CNN-RNN achieves a classification accuracy of 99.1%. Similarly, the end-to-end CNN-RNN model given re-

verberant speech achieves an accuracy of 86.9%.

Part IV

Parametric Models for Reverberant Acoustic Environments

Introduction

The previous Part of this thesis investigated the classification of reverberant acoustic environments. Chapter 6 showed that representing AIRs as FIR filters is a limiting factor to state-of-the-art classifiers. The limitations arise from the high-dimensionality and the limited number of measurements of AIRs. For the task of room classification, overcoming these limitations can benefit classifiers and allow them to learn the similarities of responses that are generated in the same room. Responses generated in the same room have shown in Chapter 5 to share characteristics related to the energy decay. At the same time, the responses exhibit individual variations that correspond to different measurement positions. These variations are strongly correlated to the structure of the early reflections. The trained room classifiers should, therefore, be able to understand these variations and how they related to the structure of the early reflections across different positions in the same room. This Part proposes a novel method for estimating parameters that form a low-dimensional representation of the early reflections in AIRs. This representation is intended to enable machines to learn properties of the reflections, without the need to process thousands of coefficients for each acoustic environment.

The rest of this Part is organised as follows:

- Chapter 7 describes how early reflections in an AIR are described by their ToA, their scale and the excitation that was used to measure the AIR. The method for estimating the above parameters is presented, using the FIR taps of measured AIR. The experiments show how accounting for frequency-dependent absorptions [7] in the room can improve the modelling and further reduce the dimensionality.

- Chapter 8 considers the frequency-dependent absorption of materials in the room. It updates the proposed reflection-parameter estimation method to include the interaction of the surfaces of materials in the room with the acoustic reflections. The Chapter also proposes a novel method for estimating the frequency-dependent sound absorption by the surfaces of materials in a room, given a single channel AIR and using DNNs. In the presented experiments, the modelled early reflections are combined with an established model for late reverberation. This experiment illustrates how the proposed modelling can be used in practice.

Chapter 7

Sparse Parametric Modelling of the Early Part of Acoustic Impulse Responses

7.1 Introduction

This Chapter proposes a novel method for estimating the ToAs and scales of reflections in the early part of AIRs and the excitation which is used for the measurement. The estimated parameters form a model which aims to accurately describe the early reflections and at the same time to reduce the dimensionality of its representation when compared to the FIR filter. The proposed model is estimated using the FIR taps of the AIR and it is able to reconstruct them, using the resulting representation. The model describes the process of sampling a sound field, which is composed of sound rays propagating in the enclosure and being reflected off its boundaries and the surface of objects. It incorporates an excitation, which is emitted by a source in order to measure the AIR, and a set of reflections modelled as superimposed delayed copies of it.

The low-dimensionality the model offers is attributed to the exploitation of the structure of the AIR, which is characterised by the sparse nature of strong early acoustic reflections. During the model fitting process, reflections are detected from the AIR coeffi-

cients, which are subsequently described by their scales and ToAs. The proposed method does not bound the ToAs to integer sample instances, it models the excitation sound and makes no assumptions about the number of overlapping reflections. These are some of the disadvantages exhibited by existing methods in the field of inverse rendering [142]–[145]. A two-stage optimisation method is employed for the parameter estimation. In the first step, an initial set of values is obtained by approximating the problem using linear regression. Regularisation is used to convey the sparse nature of the reflections to the optimisation stage of the linear regression. The second step is the fine-tuning of the model's parameter values by optimisation in local time-regions of the AIR.

The structure of this Chapter is as follows. Section 7.2 provides the formulation of the model. The method for its initialisation and subsequent parameter estimation is provided. The proposed methods for the estimation of the excitation are also presented. Section 7.6 provides experiments that investigate the performance of the modelling method for measured and simulated AIRs. A conclusion is given in Section 7.7.

7.2 Signal model

AIRs are measured by exciting an acoustic environment with an excitation signal $h_e(n)$. This excitation signal is emitted by the source, traverses the direct path to the receiver, where it is received. The excitation interacts with the enclosure to produce reflections at the boundaries and off surfaces, which are received at later sampling instances. Combining the above, a measured AIR[1] can be modelled as

$$h(n) = \sum_{i=1}^{D} \beta_i \left[h_e(n) * \frac{\sin\left[\pi(n-k_i)\right]}{\pi(n-k_i)} \right] + \nu(n), \tag{7.1}$$

for $n \in \{0, \ldots, N-1\}$. The operator $*$ indicates a convolution process and $\nu(n)$ represents the additive noise. The ToAs of the D reflections are represented by the sample indexes $k_i \in [0, \infty) \; \forall \; i \in \{1, \ldots, D\}$. In theory $D \to \infty$, however only a finite number of reflections are present in a given measurement.

[1]This can be considered as a model for the Acoustic Excitation Response rather than the Acoustic Impulse Response.

When the ToA of a reflection is an integer multiple of the sampling period and assuming frequency-independent absorptions[2], the reflection contributes to the AIR as a delayed copy of $h_e(n)$, scaled by β_i. When this is not the case and under ideal band-limiting, in addition to the delay and scaling, the excitation $h_e(n)$ is convolved with the sinc function. This concept forms the basis of the proposed method that will estimate the following parameters of (7.1):

- The ToAs of D reflections as k_i.
- The scales of D reflections as β_i.
- The parameters for the excitation $h_e(n)$.

This is a challenging non-linear problem, where the number of reflections D is unknown. The following Sections will discuss the estimation of the above parameters. The next Section discusses methods for the estimation of the excitation that was used to measure a given AIR.

7.3 Excitation estimation

Non-idealities of the AIR being modelled include the source and receiver properties such as band-limiting. These are accounted for by considering equivalent non-idealities in a corresponding excitation for the AIR. The excitation $h_e(n)$, shown in (7.1), will shape every reflection in the AIR, which will subsequently determine the accuracy of estimating the remaining parameters. Two methods for the construction of a model for the excitation are considered and are presented below.

7.3.1 Modulated Gaussian pulse

In related work, the excitation was modelled as a modulated Gaussian pulse [146]. This model is adopted here, leading to the following expression for the excitation signal

$$\hat{h}_e(n) = e^{-(\theta_{\mathrm{MGP}} n T_s)^2} \cos(2\pi f_{\mathrm{MGP}} n T_s), \tag{7.2}$$

[2]The energy absorbed by the material from incident sound is the same across all frequencies [7].

where T_s is the sampling period. The parameters θ_{MGP} and f_{MGP} are estimated from window $h_d(n)$ of the AIR containing the direct sound, modelled as

$$\hat{h}_d(n) = \hat{\beta}_d \hat{h}_e(n) * \frac{\sin\ \pi(n - \hat{k}_d)}{\pi(n - \hat{k}_d)}, \tag{7.3}$$

where k_d is the ToA of the direct sound. Assuming that the direct sound will have the highest energy, this window of length N_d is centred around the highest energy sample. Estimates for the unknowns of (7.2) are found using [147], which estimates the minimum of $\sum_n [\hat{h}_d(n) - h_d(n)]^2$.

The benefit of using a parametric model to represent the excitation signal over using samples directly extracted from the AIR is avoiding the inclusion of any overlapping reflections to the direct sound or noise in the representation. The direct use of samples from the AIR has been used in related work [144].

7.3.2 Principal components of excitations

In certain cases, a collection of AIRs is available, with the same source and receiver pair used for their measurement. This data availability is exploited in this Section to propose an alternative to the modulated Gaussian pulse approach for the modelling of the excitation, which was proposed above. To formulate this approach the following assumptions are made

- The direct-path sound is associated with the highest energy sample in an AIR.
- The source-air-receiver channel does not vary over the measurement period.
- M AIRs are available, which were measured using the same equipment.

The stipulation is that if all the above assumptions are valid and the equipment remains unchanged, then the excitation signal is generated by the same process across all measurements. The availability of M AIRs implies the availability of M observations of the process. The observed samples that correspond to the excitation however will vary across measurements. For instance, the scale of the excitation signal will depend on the distance

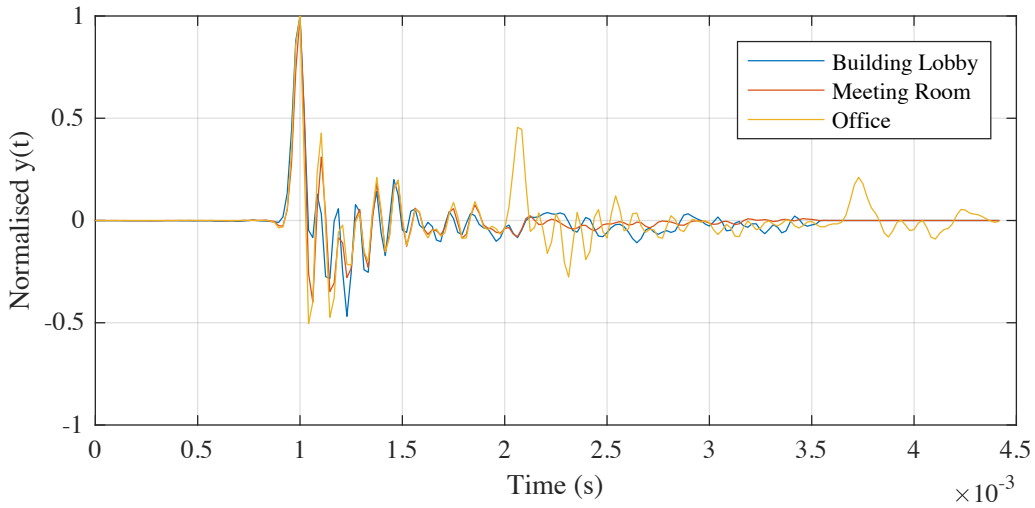


Figure 7.1: Aligned direct-sound windows using channel 1 of the Eignemike [149]. The alignment performs integer and fractional alignment of all the windows so that it minimises their amplitude mismatch and their scales are adjusted to a maximum of unity. AIRs are taken from the ACE database [11] and the measurement rooms are part of the Electrical and Electronic Engineering department of Imperial College London.

between the receiver and the source and so will its ToA [7]. The temperature of the room and the humidity are environmental factors which also affect the transmission of sounds within the environment [148]. Furthermore, the knowledge that the same equipment is used throughout the measurement of the M responses is based on the documentation of the relevant databases. The documentation is often unclear however as to whether the same *exact* equipment was used or whether the same model of equipment was used. With different devices having different characteristics, subtle differences between devices are expected.

The method proposed in this Section for estimating the excitation is based on PCA [124]. The assumption is that AIR windows containing the direct sound will involve scaled versions of the same excitation, with the possible addition of reflections and additive noise. PCA will therefore indicate high eigenvalues relating to the excitation process and the rest relating to additive noise and overlapping reflections, which are independent across the windows. The first step to estimate $\hat{h}_e(n)$ is to collect M windows of AIRs, each of N_d samples, as $\mathbf{h}_d^{(m)}$ $\forall\, m \in \{1, \ldots, M\}$. As a preprocessing step, the maximum energy samples

of all the windows are aligned. After the integer alignment, fractional alignment is done using a grid search algorithm. Collecting the resulting direct-sound windows forms the matrix

$$\mathbf{H}_d = \left[\mathbf{h}_d^{(1)}, \mathbf{h}_d^{(2)} \ldots, \mathbf{h}_d^{(M)}\right]. \tag{7.4}$$

An example of matrix $\mathbf{H}_d$ is shown in Figure 7.1. The Figure illustrates how overlapping reflections and noise are present in the windows, which should not be included in the estimate for the excitation.

After collecting samples from AIRs that correspond to the direct sound between the source and receiver, the next step is to apply PCA to $\mathbf{H}_d$. Applying PCA yields L principal components, defined by the transformation matrix $\mathbf{U}_d = [\mathbf{u}_d^{(1)}, \mathbf{u}_d^{(2)}, \ldots, \mathbf{u}_d^{(L)}]$. The choice of the number of components $\tilde{L}$ is based on the percentage of explained variance of $\mathbf{H}_d$ [150]. The excitation estimate $\hat{\mathbf{h}}_e$ for an AIR is then reconstructed using $\mathbf{h}_d$, which is the vector of its samples that containing the direct sound. The reconstruction is done using

$$\hat{\mathbf{h}}_e = \bar{\mathbf{h}}_d + \sum_{l=1}^{\tilde{L}} \left(\mathbf{h}_d^T \mathbf{u}_d^{(l)} - \bar{\mathbf{h}}_d \mathbf{u}_d^{(l)}\right) \mathbf{u}_d^{(l)}. \tag{7.5}$$

The vector $\bar{\mathbf{h}}_d$ is the average of all the vectors in $\mathbf{H}_d$. This method for estimating the excitation does not restrict it to a parametric model, as it is done by the proposed alternative, which is based on the modulated Gaussian pulse model. The process exploits the availability of multiple observations of the source-air-receiver channel to deduce the estimate. The process can be further improved in the future to account for variations in the phase repose of equipment [151] across measurements.

With methods for the estimation of the excitation, the remaining parameters which characterise individual reflections can be estimated. The next Sections describe the task of estimating the ToAs and scales of received reflections, using the provided excitation estimates.

7.4 Model initialisation

By replacing the excitation in (7.1) with the estimate derived through the methods discussed in the previous Section, the next step is to solve for the remaining parameters of the D reflections present in the AIR, which are their ToAs and scales. A two-step approach for estimating the parameters is taken, which first initialises the model and later performs a fitting process to derive the final parameter values.

7.4.1 Linear approximation

A simplification to (7.1) is made by temporarily assuming that the additive noise is negligible. A further simplification is made by reformulating the problem in order to obtain initial estimates for each of the unknowns. The simplification is to consider a fixed number of candidate reflections, at fixed ToAs. This approach was also taken in [142] but did not account for fractional reflection ToAs. In this simplified form of the problem, the only remaining unknowns are the scales of each of the candidate reflections. This transforms the estimation problem into linear regression form, given by

$$\hat{h}_R(n) = \sum_{r=1}^{M} w_r x_r(n) = \mathbf{w}^T \mathbf{x}(n) \tag{7.6}$$

$$x_r(n) = h_e(n) * \frac{\sin\ \pi\left(n - \frac{r-1}{Q}\right)}{\pi\left(n - \frac{r-1}{Q}\right)}, \tag{7.7}$$

where vector $\mathbf{w} = [w_1, \ldots, w_M]^T$ and the time-varying vector $\mathbf{x}(n) = [x_1(n), \ldots, x_M(n)]^T$. The scalar M is the number of candidate reflections considered. The ratio $Q = {}^{M}/_{N}$ is an integer and defines the number of candidate reflections per AIR coefficient. The solution to the regression problem is the set of values of the elements of $\mathbf{w}$. The unknown number of reflections is estimated from the vector as $\hat{D} = \|\mathbf{w}\|_0$, where $\|\cdot\|_0$ counts the number of non-zero elements of the vector. These elements then describe the scales β in (7.1) and their location in the vector describe the ToAs k. For the early part of the AIR, $\mathbf{w}$ is expected to be sparse as the reflection density is low.

7.4.2 Initial parameter estimation

Solving (7.6) using a LS solution [57] directly presents the following two problems, with the corresponding solutions:

- A LS solution to (7.6) will yield many non-zero values in $\mathbf{w}$ and lead to overestimates of D [142], [152]. The LASSO [57] is an appropriate alternative for the task as it promotes sparsity by penalising the L_1 norm of the regression coefficients, $\mathbf{w}$. LASSO is therefore used to find $\mathbf{w}$, which minimises the expression

$$e = \left\| h(n) - \hat{h}_R(n) \right\|_2 + \lambda \left\| \mathbf{w} \right\|_1 . \tag{7.8}$$

- Minimising the Mean Square Error (MSE) between $\hat{h}_R(n)$ and $h(n)$ will give emphasis to describing early reflections which have the highest energy. In order to account for this, the AIR $h(n)$ and the linear predictors $\mathbf{x}(n)$ of (7.6) are EDC compensated. This compensation makes reflections which appear later in time and have less energy to appear more energetic, hence also play an important part in the modelling. The compensation was originally proposed in [145] in order to better estimate the mixing time. The compensation is done here before the LASSO optimisation and involves the scaling of terms by the inverse of the EDC, given by (2.6). The regression coefficients are subject to the inverse operation after a solution is found.

The minimisation of (7.8) provides a solution for the linear regression coefficients $\mathbf{w}$. From this vector, the initial ToAs and scales are given by the vectors

$$\boldsymbol{\beta}_R = \mathbf{w} \cap \mathbb{R}_{\neq 0} \tag{7.9}$$

$$\boldsymbol{k}_R = \left\{ \frac{r-1}{Q} \ : \ w^{(r)} \neq 0 \right\}, \tag{7.10}$$

which are refined in the next Sections to form the final model parameters.

7.4.3 Adjusting regularisation

The evaluation of the initial parameter values relies on regularised regression by LASSO, that minimises (7.8). The regularisation level λ affects the level of sparsity in the result, with $\lambda = 0$ being equivalent to the LS solution. Manually adjusting λ for each AIR is impractical. Therefore the value of λ in (7.8) for the proposed model parameter estimation process is automatically adjusted. Finding the value starts by finding λ_0, the first value for which the model is not null, i.e. $\|\mathbf{w}\|_0 > 0$. LASSO is then run for

$$\lambda_\gamma = \lambda_0 \cdot 10^{-\frac{\gamma}{N_\gamma}}, \tag{7.11}$$

where $\gamma \in \{1, 2, \ldots, N_\gamma\}$. This finds solutions for an exponentially increasing regularisation level. The integer N_γ defines the search range and step size for this search. Based on the MSE value (7.8) for each γ, the λ_m value which leads to the minimum MSE ϵ_m is found. The final λ value, λ_f is chosen based on the largest γ index for which $\epsilon_\gamma < \epsilon_m + \frac{\sigma_\epsilon}{3}$. The scalar σ_ϵ describes the standard deviation of the of values of e_γ. This method of adjusting regularisation based on the standard-deviation of the error provides a trade-off between sparsity and accuracy [76].

7.5 Model fitting

The minimisation of (7.8) provides estimates for the ToAs and the scales of reflections as $\boldsymbol{\beta}_R$ and $\boldsymbol{k}_R$ respectively. The estimates are extracted from the regression coefficient vector $\mathbf{w}$. Due to the consideration of only fixed ToAs in (7.6), more than one ToA in $\boldsymbol{k}_R$ will correspond to the same reflection. The regularisation term in (7.8) also underestimates the reflection scales $\boldsymbol{\beta}_R$ [57]. In order to then better estimate the parameter values, further optimisation is performed, based on the initial estimates.

The interior-point method [153] is used to optimise reflection parameter values by minimising the MSE between the AIR reconstruction of (7.1) and windows of the original AIR. At the beginning of the process $\hat{D} = 0$, $\hat{\boldsymbol{\beta}} = \emptyset$ and $\hat{\boldsymbol{k}} = \emptyset$ are set. The optimisation of the parameters is run in a sequence of steps, each one optimising a subset of the

parameters. Each step of the optimisation considers one AIR window $h_l(n)$, which is centred at sample n_0. The optimisation starts at $n_0 = 0$ and progresses by 1 sample at each step. For each step, elements of $\mathbf{w}$ are collected to form $\mathbf{w}_0$. These are the starting point for the optimisation of the scales and ToAs of the reflections in the window, $\hat{\beta}^{(l)}$ and $\hat{\kappa}^{(l)}$ respectively. The elements $\mathbf{w}_0$ have indexes r in $\mathbf{w}$ that correspond to the one-sample region $r \in [\lfloor Q(n_0 - 0.5) \rfloor, \lfloor Q(n_0 + 0.5) \rfloor]$. The initial values for scales and ToAs are therefore given by

$$\boldsymbol{\beta}_0 = \mathbf{w}_0 \cap \mathbb{R}_{\neq 0} \tag{7.12}$$

$$\boldsymbol{k}_0 = \left\{ \frac{r-1}{Q} \ : \ w_0^{(r)} \neq 0 \right\}. \tag{7.13}$$

The optimisation for each window l minimises the MSE between the AIR window $h_l(n)$ and its reconstruction

$$\hat{h}_l(n) = \sum_{i=1}^{\hat{D}} \hat{\beta}_i \left\{ \hat{h}_e(n) * \frac{\sin\left[\pi(n - \hat{k}_i)\right]}{\pi(n - \hat{k}_i)} \right\} + \sum_{i=1}^{D_0} \left\{ \hat{\beta}_i^{(l)} \hat{h}_e(n) * \frac{\sin\left[\pi(n - \hat{k}_i^{(l)} + n_0)\right]}{\pi(n - \hat{k}_i^{(l)} + n_0)} \right\}. \tag{7.14}$$

To further promote sparsity, reflections are added to the final model if their addition scales the MSE by s. Reflections are added therefore if they substantially reduce the MSE. The accepted number of reflections is added to $\hat{D}$ and the parameters are appended to the vectors $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{k}}$, before moving to the next AIR window. Repeating the process for all windows results in a model for the AIR given by

$$\hat{h}(n) = \sum_{i=1}^{\hat{D}} \hat{\beta}_i \hat{h}_e(n) * \frac{\sin\left[\pi(n - \hat{k}_i)\right]}{\pi(n - \hat{k}_i)}. \tag{7.15}$$

The diagram in Figure 7.2 summarises the steps in the proposed method for the reflection parameter estimation from a given AIR. The later Sections will evaluate this process and show experiments that investigate the reconstruction accuracy of the model and the reduction in dimensionality it offers compared to the FIR filter representation.

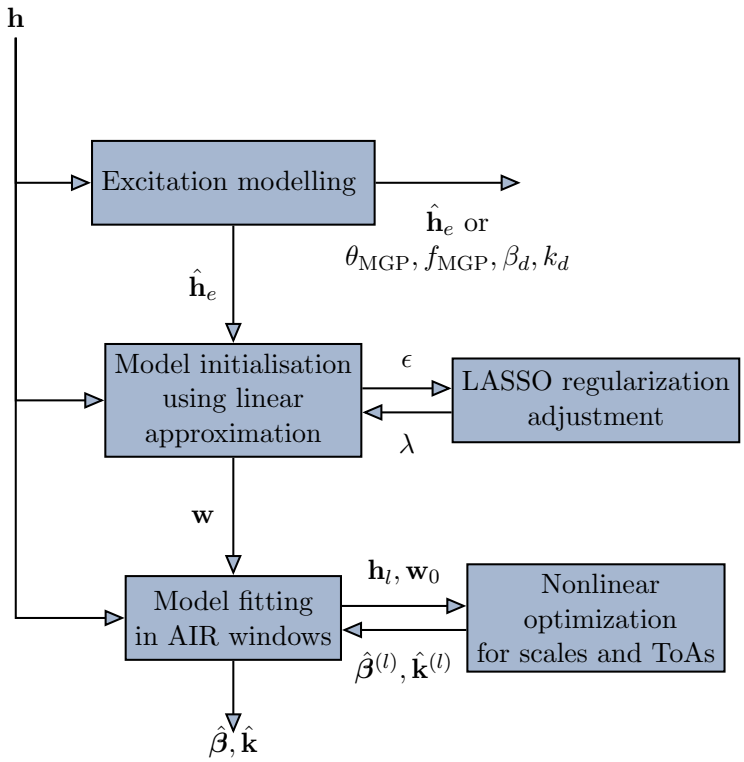
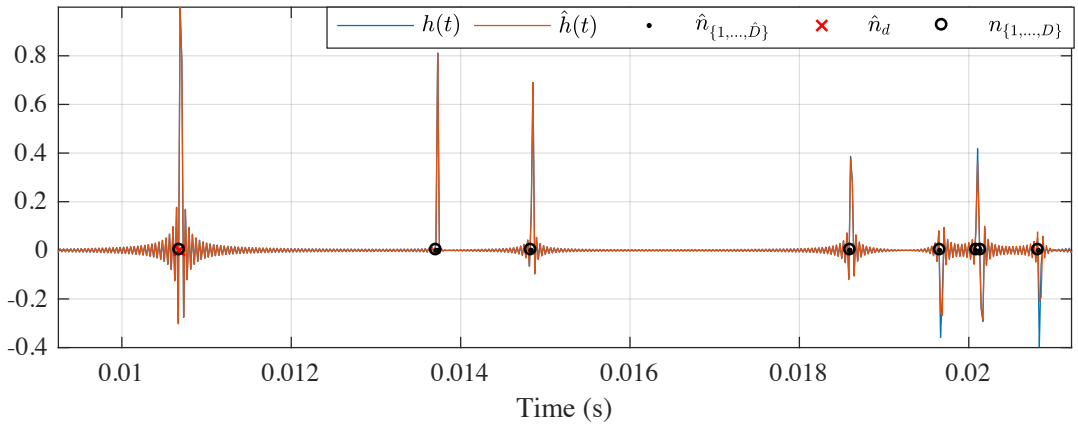


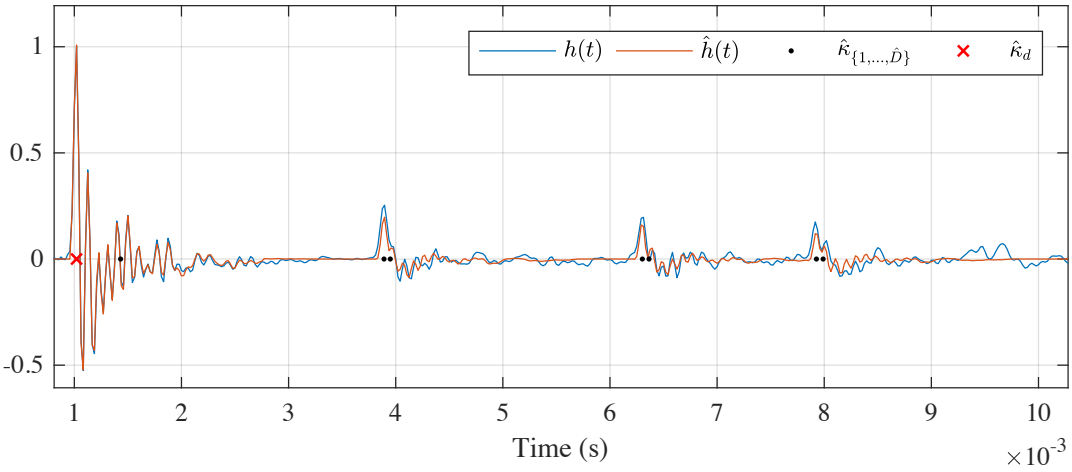
Figure 7.2: Diagram of proposed method for estimating the parameters of reflections in the early part of AIRs. The model consists of estimating the excitation used to measure the AIR and later fitting copies of that excitation to the FIR taps of the measurement.

7.6 Experiments

This Chapter so far has described the method proposed for estimating the ToAs and scales of reflections in an AIR and the excitation that was used to measure it. This Section evaluates the estimation method using simulated and measured AIRs. The experiments first provide visualisations of the result of estimating early reflection parameters using the proposed methods. The method is later evaluated using objective measures. The evaluation is based on the motivation for this work, which is the reduction in dimensionality and modelling accuracy.



(a) Estimating parameters of reflections in a simulated AIR, showing estimated and ground truth ToAs of reflections.



(b) Estimating parameters of reflections in an AIR measured in a lecture room.

Figure 7.3: Result of parameter estimation from simulated and recorded AIRs. The parameters include a model for the excitation, which is used to measure the AIR, and a ToA and scale for each of the reflections. The parameters allow for a reconstruction of the FIR filter taps of the AIR. AIRs are sampled at 48 kHz.

7.6.1 Visualising estimated parameters

Figure 7.3a shows the modelling of an AIR, simulated using [154] for a *shoe-box* room. The room dimensions are [6.27, 5.40, 2.59] m. The source and receiver are placed at a random location within the room, at least 1 m from the room's boundaries. For visual clarity, the results are shown for the first 10 ms after the arrival of the direct sound at the receiver. The ground truth ToAs of reflections, evaluated using the image method of

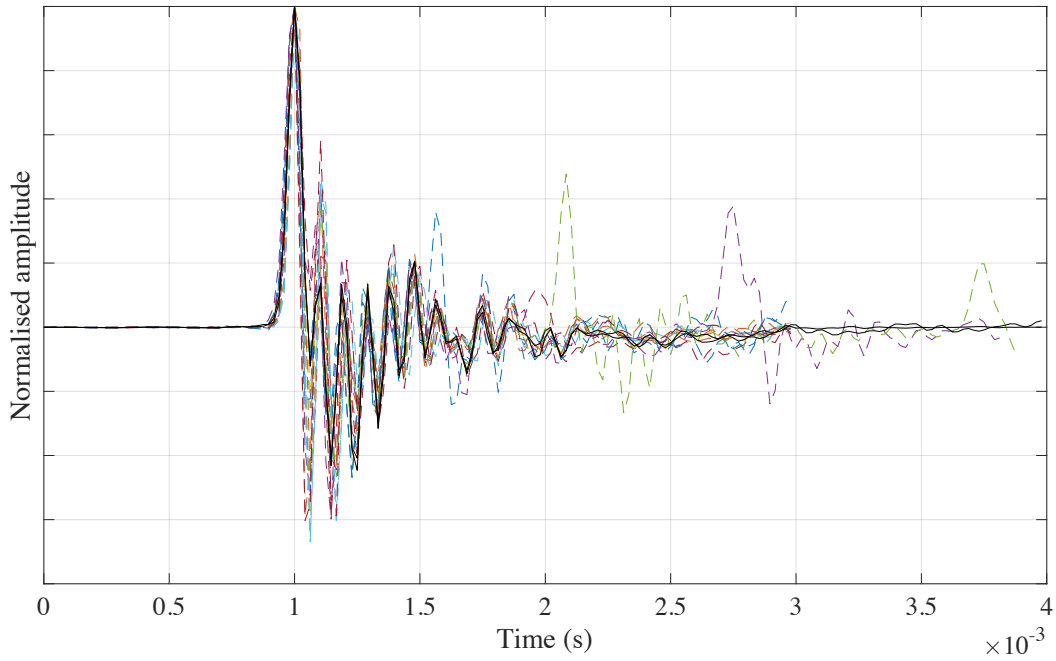


Figure 7.4: Using PCA to estimate the excitation used to measure an AIR in a lecture theatre. The estimate (black line) is the reconstruction of the window from the original AIR, which contains the direct sound, using a number of principal components which describe 95% of the variance of all the collected direct sound windows. The collected direct sound windows are gathered from AIRs measured using the same source-receiver combination, which in this case was 14 AIRs (dashed lines).

[154], are shown as κ and the estimated ToAs as $\hat{\kappa}$. Comparing $\hat{h}(t)$ to $h(t)$ shows an accurate reconstruction of the taps of the modelled AIR. The process is repeated for a measured AIR, part of the ACE Database [122], with the result shown in Figure 7.3b. The measurement took place in a lecture room with dimensions [6.9, 9.7, 3.0] m. The level of match between the reconstructed and original AIR taps is reduced when compared to the simulated AIR case of Figure 7.3a. The first taps are accurately reconstructed but as reflections arrive at later taps, the use of the same excitation signal to model them proves less effective. Also, reflections are placed close to each other, which suggests that multiple reflections can be added by the method to account for insufficient modelling power available. These points are addressed later in this discussion of Section 7.7.

The assumption made in the modelling of the reflections by the proposed method is that they are scaled and delayed copies of the excitation that was used to measure

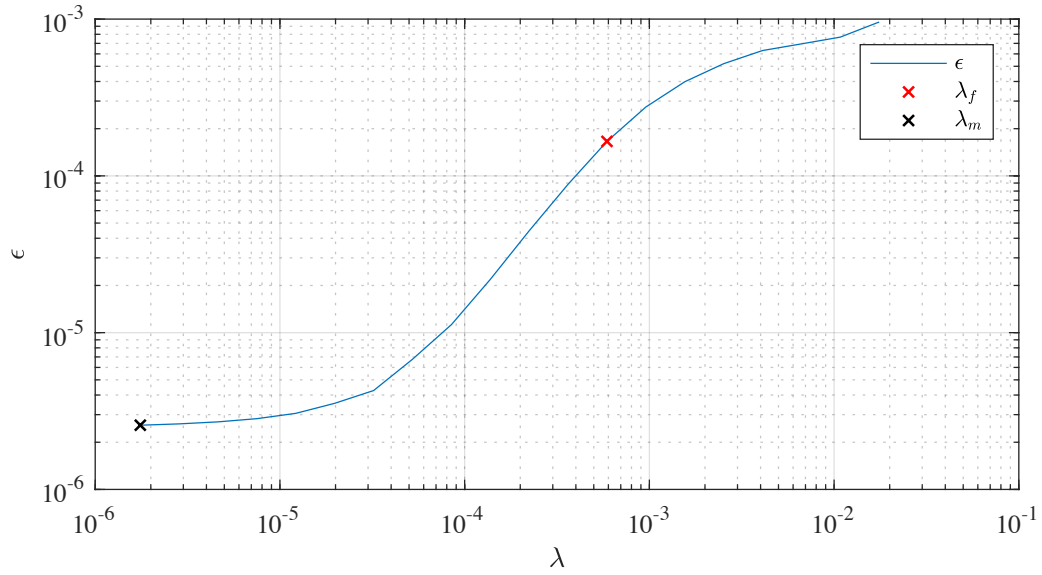


Figure 7.5: Adjusting the regularisation level of LASSO for the model initialisation process. The model uses a linear approximation to detect reflections in an AIR measured in a lecture theatre. The MSE of the reconstruction is plotted against λ, the regularisation level. The regularisation level λ_f is chosen based on its distance to the minimum MSE point λ_m and the standard deviation of the points of the curve.

the AIR. The estimation for the excitation used for the model of Figure 7.3b accurately represents the direct sound at its ToA of 1 ms. To derive this excitation, the PCA method for its estimation is used, as described in Section 7.3. For the estimation, windows are taken from 14 AIRs that were measured using the same equipment. The 14 windows are shown in Figure 7.4. The Figure shows how overlapping reflections and noise are part of the original windows and how the use of PCA excludes them from the model. 6 principal components are used for the reconstruction, explaining 97% of the variance. The estimate for the excitation is used by LASSO to estimate a set of ToAs and scales of candidate reflections. The value of (7.8), which is minimised by the LASSO regression, is shown in Figure 7.5 for values of λ. The choice of the final regularisation level λ_f is also shown. This LASSO solution estimates ToAs and scales for candidate reflections as an initialisation. These initialisations are shown in Figure 7.6. It shows detections clustered around regions of high energy. These are the values that are later optimised to provide the final solution of Figure 7.3b.

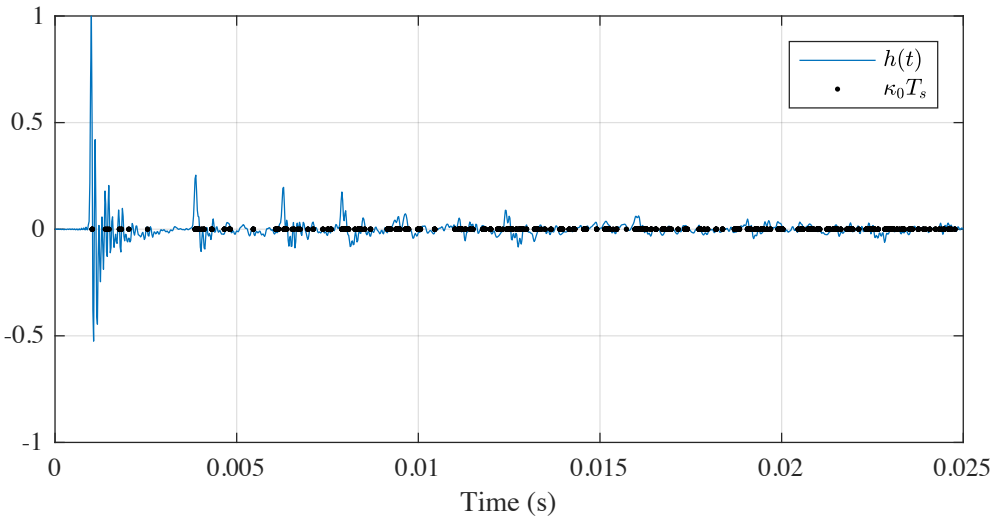


Figure 7.6: Approximating reflection ToAs in an AIR measured in a lecture room using LASSO. These approximations are used as the initialisation of the non-linear optimisation step, which models a small number of reflections that substantially reduce the MSE in reconstructing the AIR. The result of the optimisation is shown in Figure 7.3b.

While visualisations are useful in understanding the process of estimating reflection parameters, as proposed in this Chapter they do not offer objective insight into its performance. This is done in the next Section.

7.6.2 Evaluation using objective measures

The experiments above visualised parameters of reflections that are estimated using the method proposed in this Chapter. Objective evaluation of the modelling offered by these parameters will be given in the following experiments through the use of the normalised error and the dimensionality reduction values. These are respectively denoted by ψ and ζ. The normalised error is defined as

$$\psi = \frac{\|h(n) - \psi_{\text{scale}}\, \hat{h}(n)\|_2}{\|h(n)\|_2} \times 100\%, \tag{7.16}$$

which expresses the residual error as a percentage of the overall AIR energy. The scalar ψ_{scale} is the LS solution to $h(n) = \psi_{\text{scale}}\, \hat{h}(n)$, added to normalise the levels of the estimate

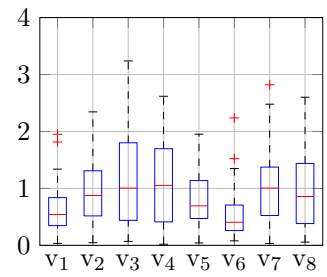
and the original AIR. The dimensionality reduction is evaluated as the percentage

$$\zeta = \left(1 - \frac{N_{\text{param.}}}{N}\right) \times 100\%, \tag{7.17}$$

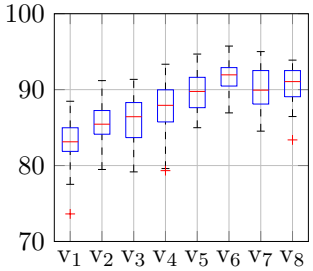
where $N_{\text{param.}}$ is the number of parameters estimated by the proposed method and N the number of taps required for the FIR filter representation. The parameters estimated are the ToAs and scales of $\hat{D}$ reflections and either 4 parameters to describe the direct excitation model of (7.2), or the N_d samples of the PCA extracted $\hat{h}_e$ from (7.5). For the model fitting process (see Section 7.5), s was set to 0.90, τ_e to 0.5 ms, Q to 20 samples, N_d to 59 samples and N_γ to 25. This provided a trade-off between sparse solutions and modelling accuracy.

Two experiments are conducted to evaluate the method, looking at simulated and measured AIRs. In the first, 480 AIRs are simulated at a sampling frequency of 16 kHz, using [155]. This involves 15 rooms of different dimensions with a single source and 32 receivers in each room. Source and receivers are randomly placed, at least 1 m away from the boundaries. The model is fitted to the first 24 ms after the arrival of the direct sound for each. This is defined as the mixing time in [31]. Rooms are split into 8 groups based on their volume. The normalised error and dimensionality reduction values offered by the estimated parameters per group are shown in Figure 7.7. For the second experiment, the task is to model 42 AIRs measured in 7 rooms provided in the ACE database [122]. The AIRs are downsampled to a sampling frequency of 16 kHz. This corresponds to 2 measurements per room, with the receiver positions varying between the two. The normalised error and dimensionality reduction values offered by the estimated parameters per room are shown in Figure 7.8. Higher indexes for a specific room type indicate a higher room volume, i.e. Lecture Room 2 has a higher volume than Lecture Room 1 and so on. For these experiments, the excitation was modelled by fitting the modulated Gaussian pulse model of Section 7.3.1.

The following Section offers a discussion which analyses the results of the experiments above. The benefits of using the proposed method for modelling reflections in the early part of AIRs are illustrated and points are made as to the cases where the model performs



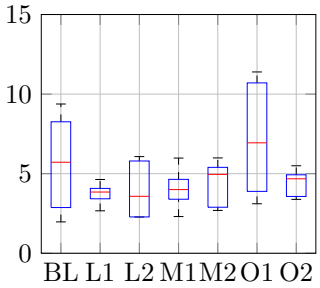
(a) Normalised Error (%)



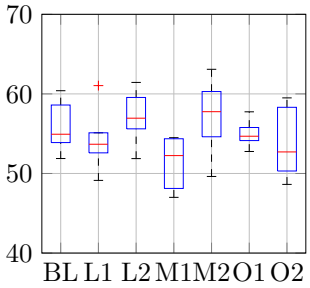
(b) Dimensionality reduction (%)

Group Legend: v1: 15-24 m^3, v2: 25-34 m^3, v3: 35-44 m^3, v4: 55-64 m^3, v5: 65-74 m^3, v6: 85-94 m^3, v7: 95-104 m^3, v8: 105-114 m^3

Figure 7.7: Modelling accuracy and dimensionality reduction offered by modelling reflections in simulated AIRs. 480 AIRs are simulated in 15 rooms. Rooms are grouped in terms of volume with increasing indexes indicating a larger room volume.



(a) Normalised Error (%)



(b) Dimensionality reduction (%)

Room Legend : BL:Building Lobby, L1: Lecture Room 1, L2: Lecture Room 2, M1: Meeting Room 1, M2: Meeting Room 2, O1: Office 1, O2: Office. Room Volume: BL: 72 m^3, L1: 200 m^3, L2: 360 m^3, M1: 92 m^3, M2: 250 m^3, O1: 47 m^3, O2: 48 m^3

Figure 7.8: Modelling accuracy and dimensionality reduction offered by modelling reflections in measured AIRs. 42 AIRs are modelled, which were measured in 7 rooms.

best. Links between the properties of the rooms and the performance of the modelling are established and presented.

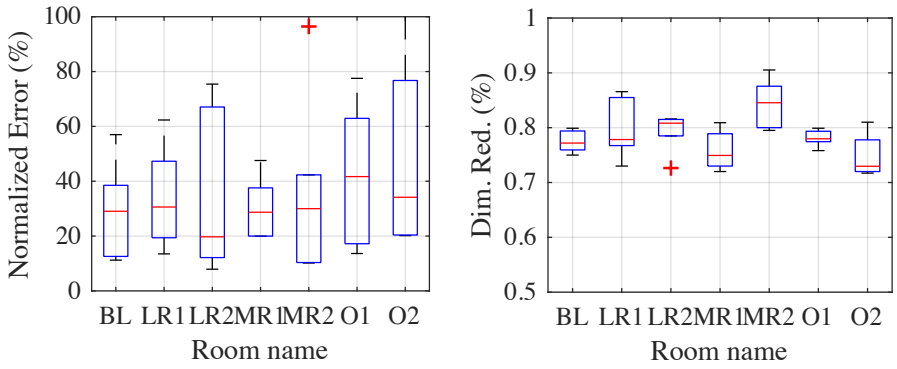
7.7 Discussion and conclusion

The result of modelling 480 simulated AIRs is shown in Figure 7.7. The normalised error values indicate a constant modelling accuracy across room volume groups for the ideal *shoe-box* scenario, with the error never exceeding 3.5%. This shows invariance in terms of the modelling accuracy with regards to the size of simulated rooms. The dimensionality

reduction results show that more parameters are required to model smaller rooms. For the smallest room-volume group, the median dimensionality reduction is 83%, whereas for larger rooms it reaches 92%. This increase in dimensionality for decreasing sizes of ideal rooms is due to the more rapidly increasing reflection density in smaller enclosures [8].

The results of modelling 42 measured AIRs in real rooms provide insight into the capabilities of the proposed method in realistic scenarios. The normalised error values in Figure 7.8 now reach 11%, indicating the increased challenges in modelling real AIRs. The dimensionality reduction is lower overall compared to the case of simulated AIRs, with values ranging between 47–63%. For measured AIRs, the model has to include further parameters to account for reflections from objects other than the enclosure's boundaries, which are not present in the ideal *shoe-box* room scenario. For instance, reflections from the surfaces of furniture present in the room are part of the measured data, which is not the case for the simulations. The increased number of surfaces in real rooms is not the only difference. The surfaces of objects in realistic acoustic environments absorb sound at different levels at different frequencies. This is referred to as frequency-dependent absorption of sound [8] and its characteristics depend on the material of the surfaces. This means that reflections are not perfect copies of the excitation but are stretched and altered versions of it. This effect is studied in detail in Chapter 8. Ambient and sensor noise in AIR measurements is expected to also impact the residual error. These factors make the reflection parameter estimation task more challenging, which affects both the error and the dimensionality reduction values.

In contrast to simulated AIRs, larger real rooms do not consistently lead to a higher dimensionality reduction. Nevertheless, higher reductions in dimensionality are still shown between rooms of the same type as their volume increases. This indicates that volume is again a factor to be considered in terms of the model's dimensionality. The normalised error in Figure 7.8 shows high variability for specific rooms across measurements. The two rooms with the highest variability are the Building Lobby and Office 1. Investigating further indicates that the measurement position is related to the level of errors. For both rooms, the AIRs with the highest modelling error involved the receiver being placed closest



(a) Normalised Error (b) Dimensionality reduction

Figure 7.9: Accuracy and dimensionality reduction gained by a method that uses samples from the AIR to represent the excitation signal and does not account for fractional ToAs for 42 measured AIRs part of the ACE database.

to a room wall. The opposite was true for receiver locations closest to the middle of the room. AIRs closer to the room boundaries are therefore more challenging to model. This is attributed to the fact that reflection spacing will be smaller, making the modelling task more difficult.

To illustrate the benefits of the proposed method over alternatives in the modelling of measured AIRs, Figure 7.9 is provided. It shows the normalised-error and the dimensionality reduction offered by a method that is equivalent to the proposed one but, which does not account for fractional reflection ToAs and models the excitation signal as samples directly taken from the AIR. Comparing the results to those of the proposed method in Figure 7.8, shows that the alternative results in error values up to 10 times higher than the proposed one. The ability of the proposed method to capture fractional ToAs results in a number of reflections being accurately modelled, leading to an accurate reconstruction of the AIR. The alternative models fewer reflections, bound at integer ToAs, leading to a smaller parameter set and larger dimensionality reduction values.

In conclusion, this Chapter presented a novel approach for the estimation of the parameters of early reflections in an AIR and the excitation used to measure it. The parameters are used to form a low dimensional parametric model, which describes sparse acoustic reflections by their ToAs and scales. The model can reconstruct the FIR taps of

the AIR by delaying and scaling the excitation. The motivation for this work is to find an accurate and low dimensional representation for the early reflections in AIRs that enables machines to learn properties of the reverberation effect. The estimated reflection ToAs by the proposed method also find direct applications in the estimation of room geometry [143] and the mixing time [145]. In experiments involving simulated and measured AIRs, the proposed method reduced their dimensionality by more than 90% and 60% respectively. The corresponding AIR coefficient reconstruction normalised-error did not exceed 3.2% for simulated and 14.0% for measured AIRs. The next Chapter improves the proposed method, by accounting for frequency-dependent absorptions [7], in order to increase its modelling power.

Chapter 8

Material-aware Modelling of Reflections in AIRs

This Chapter proposes a novel method for estimating the level of frequency-dependent absorptions by the surfaces of materials in an acoustic environment. This Chapter also proposes a method for improving the modelling of early reflections by considering their interaction with the surfaces of the materials present in the environment. The method is based on the sparse and low-dimensional modelling presented in Chapter 7. The estimation of the parameters is done using a single-channel AIR measured in the environment.

While extracting information such as the dimensions of a physical room is typically feasible [101], the same does not apply to the measurement of the frequency-dependent absorption of the surfaces in it. Two common methods for this measurement are outlined in ISO 354 [156] and ISO 10534 [157]. Both methods require the extraction of samples of materials from the environment, which can be impractical and undesirable. The method proposed in this Chapter estimates the frequency-dependent absorptions of the materials present in the room, using one AIR measured in the room. The method is based on a detector DNN, that is trained using *a priori* knowledge of the properties of materials. The network detects the presence of materials with specific absorption properties. The proposed method finds many applications, such as room-geometry estimation [98] and inverse rendering [1]. In this Chapter, it is used for the modelling of specular acoustic

reflections arriving at the receiver. The parameters of the reflections provide a parametric model for the early part of AIRs. The resulting model, having many fewer parameters than an FIR filter, can reconstruct the original high-dimensional FIR filter. The motivation for this work is to find a low-dimensional and accurate representation of the early reflections that enables machines to better understand properties of the reverberation effect.

This rest of the Chapter is organised as follows: Section 8.1 presents the notation used to model the interaction of sounds with the surfaces of materials in the environment. Section 8.2 presents the proposed method for estimating the frequency-dependent adsorptions of materials in the environment and Section 8.3 analyses the data used to train the method. The estimated absorption values are used in Section 8.4 to propose a method for estimating the parameters of early reflections and their interaction with the surfaces of materials in the room. The accuracy of the estimated frequency-dependent adsorptions in an environment is evaluated in the experiments of Section 8.5 and subsequent experiments in Section 8.6 evaluate the proposed reflection parameter estimation method. The experiments include the use of the estimated parameters in combination with established methods for representing the late reverberation, in order to create a reconstruction of entire AIRs. A discussion and conclusion are given in Section 8.7.

8.1 Modelling sound absorption in acoustic environments

The background material in Section 2.3.1 introduced the theory of sound absorption by materials. Recalling from this introduction, the level of absorption for a specific material is typically represented by a coefficient a, which is the proportion of incident sound energy it absorbs. A coefficient of $a = 1$ indicates that no energy is reflected back into the room.

In order to represent the absorption process for a material θ as a filter, an FIR filter $\phi_\theta(n)$ is defined with Discrete Fourier Transform (DFT) as $\Phi_\theta(u)$. n indicates the sample index and u the DFT bin. Any incident sound to the material is therefore filtered by $\Phi_\theta(u)$ before being reflected back into the room. Sound absorption coefficients for a set of frequencies are given in a table form in the literature [8]. These frequencies refer to

the geometric centres of octave bands. Absorption coefficients only provide information with regards to $|\Phi_\theta(u)|$ and not the phase response [8]. The magnitude response of $\Phi_\theta(u)$ can therefore be constructed by the tables in a way that it scales the energy of the input, according to the energy absorption coefficient by the material

$$\alpha(u) = 1 - |R(u)|^2. \tag{8.1}$$

at each frequency bin. $R(u)$ represents the changes in amplitude and phase of sounds incident to the material [7].

For a room, the collection of such filters is denoted with matrix $\mathbf{\Phi}$

$$\mathbf{\Phi} = [\phi_1(n), \phi_2(n), \ldots, \phi_\Theta(n)]^T \tag{8.2}$$

where $\theta = \{1, \ldots, \Theta\}$ denotes present materials. The next Sections looks at how these filters are used to improve the modelling of early reflections, compared to the method of Chapter 7.

8.1.1 Modelling the absorption process

Chapter 7 introduced the following model for early reflections at the receiver:

$$h(n) = \sum_{i=1}^{D} \beta_i \left[h_e(n) * \frac{\sin[\pi(n - k_i)]}{\pi(n - k_i)} \right] + \nu(n). \tag{8.3}$$

The model describes the AIR between the source of the excitation $h_e(n)$ and the receiver, as the superposition of D reflections. Each reflection is modelled as a copy of the excitation delayed by k_i samples and scaled by a factor β_i. The value of k_i is in general non-integer. Additive noise is represented by $\nu(n)$. The model was a simplification of the physical process of reverberation. Frequency-dependent absorption of acoustic energy from materials in the room was not accounted for. To account for them, filters $g_i(n)$ are incorporated into (7.1), which describe the collective effect of the absorption process on the excitation signal $h_e(n)$ as it is reflected off a number of surfaces along its path. (7.1) is therefore rewritten as

$$h(n) = \sum_{i=1}^{D} \beta_i \left[h_e(n) * g_i(n) * \frac{\sin\left[\pi(n - k_i)\right]}{\pi(n - k_i)} \right] + \nu(n). \tag{8.4}$$

The filters $g_i(n)$ are defined in the discrete frequency domain as follows

$$G_i(u) = \mathfrak{F}\{g_i(n)\} = \prod_{\theta=0}^{\Theta-1} \left[\Phi_\theta(u)\right]^{r_{\theta,i}}, \tag{8.5}$$

for a reflection i, a frequency bin u, for materials $\theta \in \{1, \ldots, \Theta\}$ and the DFT operation $\mathfrak{F}\{\cdot\}$. Θ is the number of materials present in the enclosure and the positive integer $r_{\theta,i}$ is the number of times reflection i was incident on material θ. This represents the successive filtering of the incident sound by the surfaces of the materials. Using the properties of the DFT and assuming that the material filters are Linear Time Invariant (LTI), equation (8.5) describes the overall process.

The link between the physical process and the factor β_i has changed from the model of Chapter 7. The factor now accounts for the air absorption α_i^{air}, the propagation loss and also for the collective effect of phase inversions and scattering at each incidence for each reflection. Therefore, for reflection i of order Ξ the scaling term is described by

$$\beta_i = \frac{\lambda_{i,\xi}(-1)^p}{r} \sqrt{1 - \alpha_i^{\text{air}}} \prod_{\xi=0}^{\Xi-1} \gamma_{i,\xi}, \tag{8.6}$$

where $\lambda_{i,\xi}$ is a positive scalar representing the uncertainty in the modelling of material filters $\phi_\theta(n)$ and the variations due to the angle of incidence, p is a binary variable to account for possible phase inversions, r is the distance travelled by the reflection and $\gamma_{i,\xi}$ is the scaling due to the scattering loss at reflection order ξ. The scaling of the incident sound amplitude by the inverse of the distance assumes a point omnidirectional source [7].

With the frequency-dependent absorptions incorporated into the reflection model, the next task is to define methods to estimate the reflection parameters. Similarly to Chapter 7, this estimation will be done from the FIR taps of AIRs.

8.1.2 Inverting the model

Given an AIR, this Chapter offers a novel method for estimating the parameters of early reflections given by (8.4). The parameters are estimated in order to minimise the error between a reconstruction $\hat{h}(n)$ and the actual AIR $h(n)$. The estimated parameters are the following:

- The excitation signal $h_e(n)$, given by vector $\mathbf{h}_e$.
- The ToAs of D reflections k_i, given by vector $\boldsymbol{\kappa}$.
- The scales of D reflections β_i, given by vector $\boldsymbol{\beta}$.
- The absorption filters of Θ materials $\hat{\phi}_\theta(n)$, forming the rows of $\boldsymbol{\Phi}$.
- The integer number of times $r_{\theta,i}$, a reflection i was reflected off a surface of material θ. The collection of the integers for all reflections and materials forms matrix $\mathbf{R}$, with rows corresponding to reflections and columns corresponding to materials.

The task in this Chapter is to solve

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\kappa}}, \hat{\boldsymbol{\Phi}}, \hat{\mathbf{R}}) = \underset{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\kappa}}, \hat{\boldsymbol{\Phi}}, \hat{\mathbf{R}}}{\operatorname{argmin}} \epsilon(\mathbf{h}, \hat{\mathbf{h}}\ ;\ \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\kappa}}, \hat{\boldsymbol{\Phi}}, \hat{\mathbf{R}}), \tag{8.7}$$

where $\epsilon(\mathbf{h}, \hat{\mathbf{h}})$ a function that measures the reconstruction error. With estimates of all parameters except the material filters and their interaction with the excitation being provided from the method proposed in Chapter 7, the rest of this Chapter focuses on methods for the estimation of the matrices $\boldsymbol{\Phi}$ and $\mathbf{R}$.

8.2 Detecting material types present in an acoustic environment

This Section discusses the estimation of the filters $\phi_\theta(n)$ that form matrix $\boldsymbol{\Phi}$ from an AIR. Following this, the subsequent steps that estimate the interaction of the acoustic reflections with the materials described by the filters are presented.

Recalling from the literature review of Section 4.1.1, the following two categories of existing methods have been proposed for the estimation of the frequency-dependent absorptions of materials in acoustic environments.

1. Estimate the material absorptions, for a known room geometry and with access to measured AIRs [1].

2. Estimate the material absorptions, having both visual access to the environment through camera images and to AIRs measured in the environment. This diversity of information allows for the blind estimation of the absorption coefficients using as initial estimates values of coefficients taken from lookup tables [105].

The ideal solution, in the case of this Chapter, would be to perform the same type of blind estimation as the second category but using only measured AIRs. One approach for this would be to blindly estimate the coefficients directly from the AIRs. The task of blind estimation of the material filters is computationally high and would inevitably need to make several assumptions about the acoustic environment in order to be solved. The alternative solution used in this Chapter is based on using *a priori* knowledge of material-absorption properties. This knowledge is used to train machines, able to detect patterns in the AIRs, associated with these absorption properties.

An approach for estimating material-absorption filters in an acoustic environment is proposed in the following Sections, which is based on learning from *a priori* knowledge of materials. The reliability of the estimates is evaluated and their usefulness in the representation of early reflections is studied.

8.2.1 Detection of known absorption types

This Section describes a method for estimating the frequency-dependent absorption of sound by the surfaces of materials in a room. The method uses values from tables in the literature for training a DNN detector for the presence of materials, based on their absorption properties. The detector estimates the probability of presence of a given material in the acoustic environment, given the FIR taps of an AIR measured in the environment.

The proposed method starts with collecting data from a table of frequency-dependent absorptions values for a set of materials. These are typically given in octave bands for frequencies between 63 Hz and 8 kHz. This gives 8 energy absorption coefficients at 63, 125, 250, 500, 1000, 2000, 4000 and 8000 Hz, for each material θ. Packing these 8 values together forms column vector $\mathbf{a}_\theta$. Values for Θ_{tot} materials are collected from tables that form the matrix of absorption coefficients $\mathbf{A}_{\text{tot}} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{\Theta\text{tot}}]^T$, with $0 < a < 1 \; \forall \; a \in \mathbf{a} \; \forall \; \mathbf{a} \in \mathbf{A}_{\text{tot}}$.

The frequency-dependent absorptions of the materials present in an environment is $\mathbf{A}$. This Chapter constructs this matrix by choosing the appropriate rows of the matrix of known absorptions $\mathbf{A}_{\text{tot}}$, indicating only the materials that are present in the environment. The aim is therefore to design a model d, able to perform the task

$$\mathbf{A} = d(\mathbf{h}, \mathbf{A}_{\text{tot}}). \tag{8.8}$$

This will detect the materials which are present in the environment and associate them with their known sound absorption properties from the literature.

This treatment of the problem as a detection task instead of an estimation task simplifies the problem, as the filter coefficients can be drawn from predesigned material-filterbanks. It also allows for the use of state-of-the-art detector networks from the literature. The DNN detector used in this work is detailed in the next Section.

8.2.2 DNNs as detector models

For the task of estimating the frequency-dependent absorptions in an environment, the method proposed in this Chapter starts with detecting the presence of different materials. The detected materials are associated with their known sound absorption properties from the literature, which are later used to model the frequency-dependent absorptions in the room.

For the choice of the detector mechanism for the presence of materials in the room, the focus is on the use of DNNs. DNNs have shown great success in the field of SED, as

illustrated in the recent DCASE challenges [52], [110]. Approaches proposed [107] have shown that an appropriately formed input and architecture can perform well in the task of detecting individual sound events. Also, [158] has illustrated the success of DNNs in the detection of rare sound events. The success of DNNs in SED motivates their consideration for the detection of materials in this Chapter.

The aims of SED are translated to the motivation of this work, by considering reflections under the general concept of acoustic events. Therefore, different types of absorptions, giving specific characteristics to each reflection, will yield different sound events. Following the above reasoning, a DNN is trained to estimate the probability of occurrence of an event of a certain category. In this case, this indicates the probability that a certain type of absorption took place, given the entire AIR. Going back to the formulation of this approach, the model therefore estimates the probabilities

$$p(\mathbf{a}_\theta \in \mathbf{A}|\mathbf{h}) \; \forall \; \theta \in \{1, \ldots, \Theta_{\text{tot}}\}\,. \tag{8.9}$$

To detect whether a material θ is present in the environment, a threshold ζ_θ is applied to the posterior, giving the detector

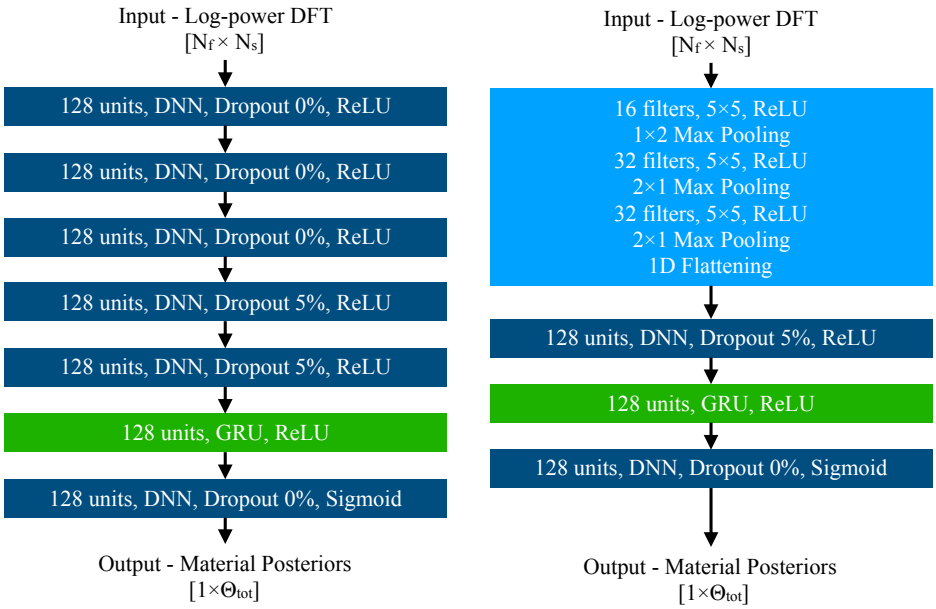
$$a_\theta \in \mathbf{A} = \begin{cases} \text{True}, & \text{if } p(\mathbf{a}_\theta \in \mathbf{A}|\mathbf{h}) > \zeta_\theta \\ \text{False}, & \text{otherwise} \end{cases}. \tag{8.10}$$

The typical choice for the detection threshold is 0.5 [63]. Another approach is to choose a value that provides a balance between Type I and Type II errors [57].

8.2.3 Proposed detector model architectures

Borrowing insight from the field of SED, the use of convolutional and recurrent layers is investigated for the material detector. Two architectures are studied, which are both evaluated in the experiments in later Sections in terms of their detection accuracy.

Combining convolutional and recurrent layers was investigated in [66], which explains its benefits for the processing of audio using the resulting CNN-RNN. In Chapter 6 of this

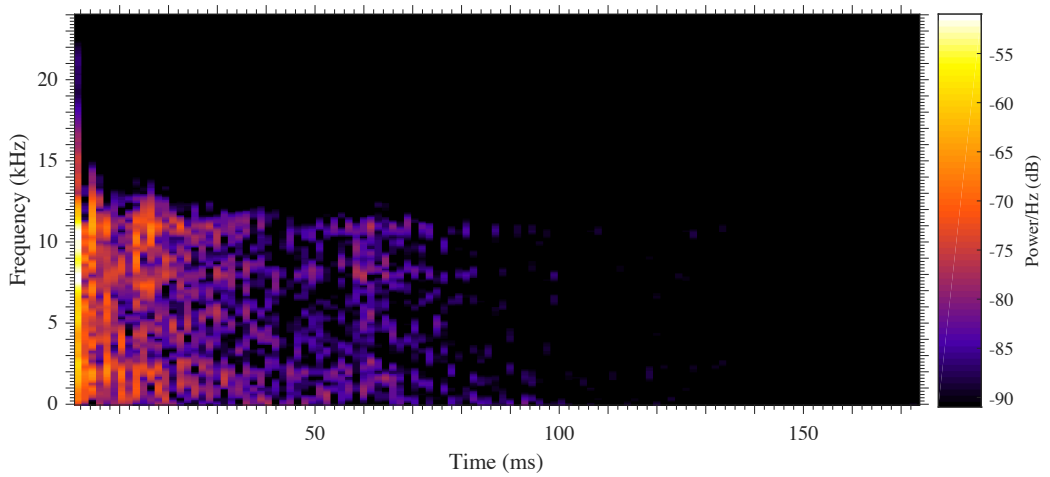


(a) FF-RNN (b) CNN-RNN

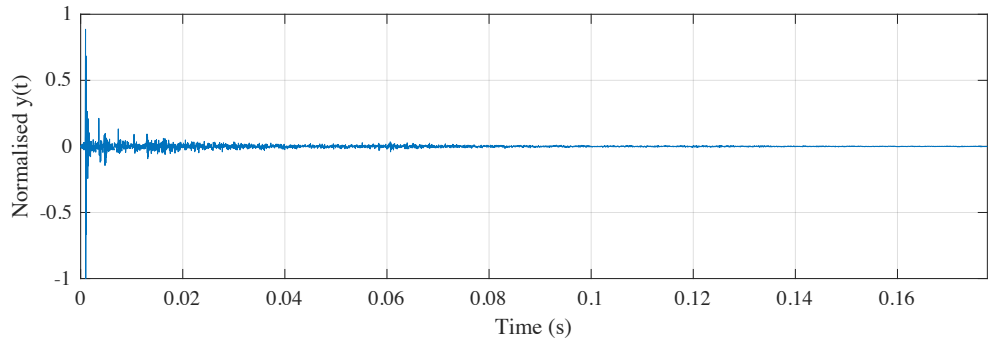
Figure 8.1: Proposed models for the detection of materials present in a reverberant acoustic environment, based on their frequency-dependent sound absorption characteristics.

thesis, the architecture was very successful in processing AIRs to categorise individual rooms. Convolutional layers learn high-level features in a shift-invariant manner [159] and subsequent Max Pooling layers remove variations in the frequency domain. These properties are very beneficial for the task at hand. Furthermore, recurrent layers learn long- and short-term temporal patterns in the data and retain that information to provide meaningful outputs with respect to the task. The recurrent layers are designed using GRU cells [65]. GRUs have shown evidence of faster convergence and increased accuracy in certain scenarios [160], [161], when compared to other recurrent cells. The choice of adding FF layers before the recurrent layer serves the twofold purpose of dimensionality reduction [66] and generalisation by incorporating dropout strategies [46] in order to avoid overfitting in such a complex structure. For comparison, the use of a FF-RNN architecture is also investigated.

The diagrams in Figure 8.1 show the 2 models investigated for the task of detecting the presence of materials in acoustic environments using an AIR as the input. The input AIR is segmented into frames of duration 3 ms and a 1.5 ms overlap. This provides fine



(a) Log-power representation of the AIR



(b) AIR samples

Figure 8.2: Log-power spectrogram extracted from an AIR, measured in a meeting room (ACE database[11]), for use with the proposed models. The spectrogram is the input to the DNNs, which detect the presence of materials with different sound absorption properties in the acoustic environment.

temporal resolution in order to analyse recordings at the reflection level and still a large enough number of samples to maintain significant spectral resolution. The log-power in the discrete frequency domain is presented for each frame at the input, similar to [6]. An example of the resulting input to the networks for a measured AIR is given in Figure 8.2.

Training the DNNs as discussed in this Chapter enables the detection of materials present in the acoustic environment, based on their sound absorption properties. The detected materials will provide information about the frequency-dependent absorption of sound in the environment. This information will be used to improve the modelling of

the early reflections. To train these networks, a set of data is needed that is labelled with ground truth information about the materials present in the room. The next Section describes the process of collecting such data and the challenges in doing so.

8.3 Analysis of data for material sound absorption

This Section discusses the source of the *a priori* information about material sound-absorption properties, which is used to create the training data for the detector DNN discussed in the previous section.

Knowledge of the frequency-dependent absorption of sound by materials is available in the literature as acousticians use this information as a reference for auralisation experiments and in the design of auditoria. Material manufacturers, especially of specialised materials, make this information available to facilitate the efforts of acousticians. The software package Odeon[1] is a modelling software that combines such information with acoustic models to create auralisations. Given the popularity of the software and the fact that a number of manufacturers release their data in a compatible format with it, the absorption data that is available on the software's page is used for the training of the detector DNN[2]. The provided data with the software are called *reference* data, from which 163 materials are taken from 15 categories. The distribution of these materials across the different categories is given in Figure 8.3. They span a wide range of material types with diverse absorption coefficients. The original sources of this information are from the literature and also include samples directly provided from manufacturers, making this list diverse and representative. Expanding this list in the future simply involves downloading data from the website of manufacturers and merging them with the *reference* list. The absorption values in the list are encoded in the format described in Section 8.2.1, with 8 absorption coefficients provided for the 1-octave bands in the range 125 Hz to 8 kHz.

[1]Software homepage: `https://odeon.dk/`

[2]The list is freely and publicly available in an electronic format at the time of writing of this thesis here: `https://odeon.dk/sites/all/themes/odeon/images/Materials/Material.Li8`

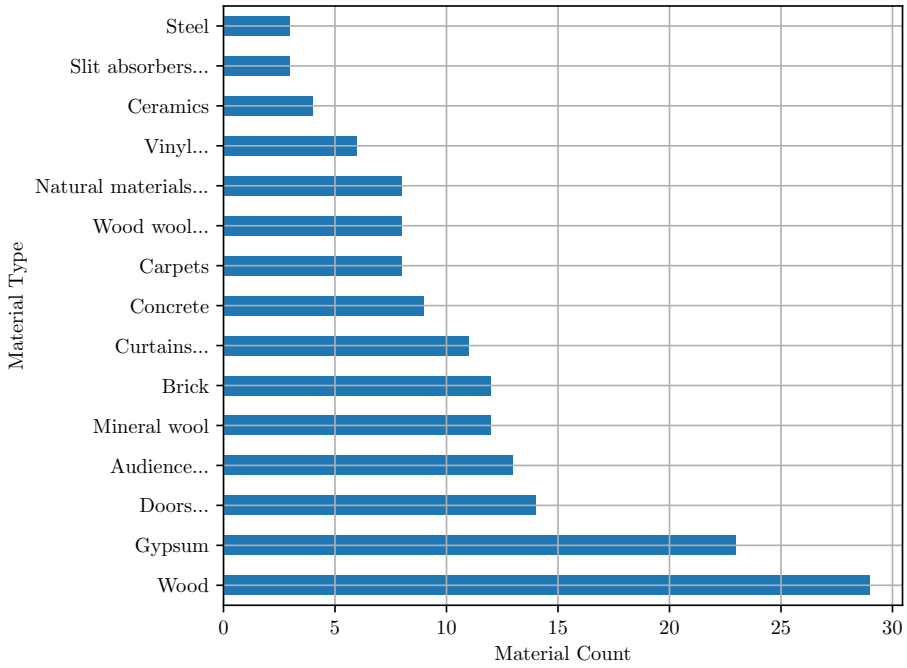


Figure 8.3: Distribution of *reference* materials across different types as, provided with the Odeon modelling software.

8.3.1 Ambiguities in the problem

Returning back to the motivation of this data harvesting, the aim is to train models that enable the detection of the presence of material θ in a room. A material is characterised by its frequency-dependent absorptions as a vector $\mathbf{a}_\theta$. Inspecting the values of reference materials in terms of their absorption coefficients shows that a number of them share the exact same coefficient values. This is not unexpected as different material compositions can lead to different visual appearances or different textures, however their sound absorption properties can remain the same [104]. The task of distinguishing between materials as they are listed in raw tables of absorptions is therefore impossible, due to the inherent ambiguities in the problem. Only approaches that combine audio and visual information can lead to a partial resolution of these ambiguities [105]. The interest of this work however is to perform this task using only a single AIR.

Given the above sources of ambiguities, the objective of material detectors from AIRs

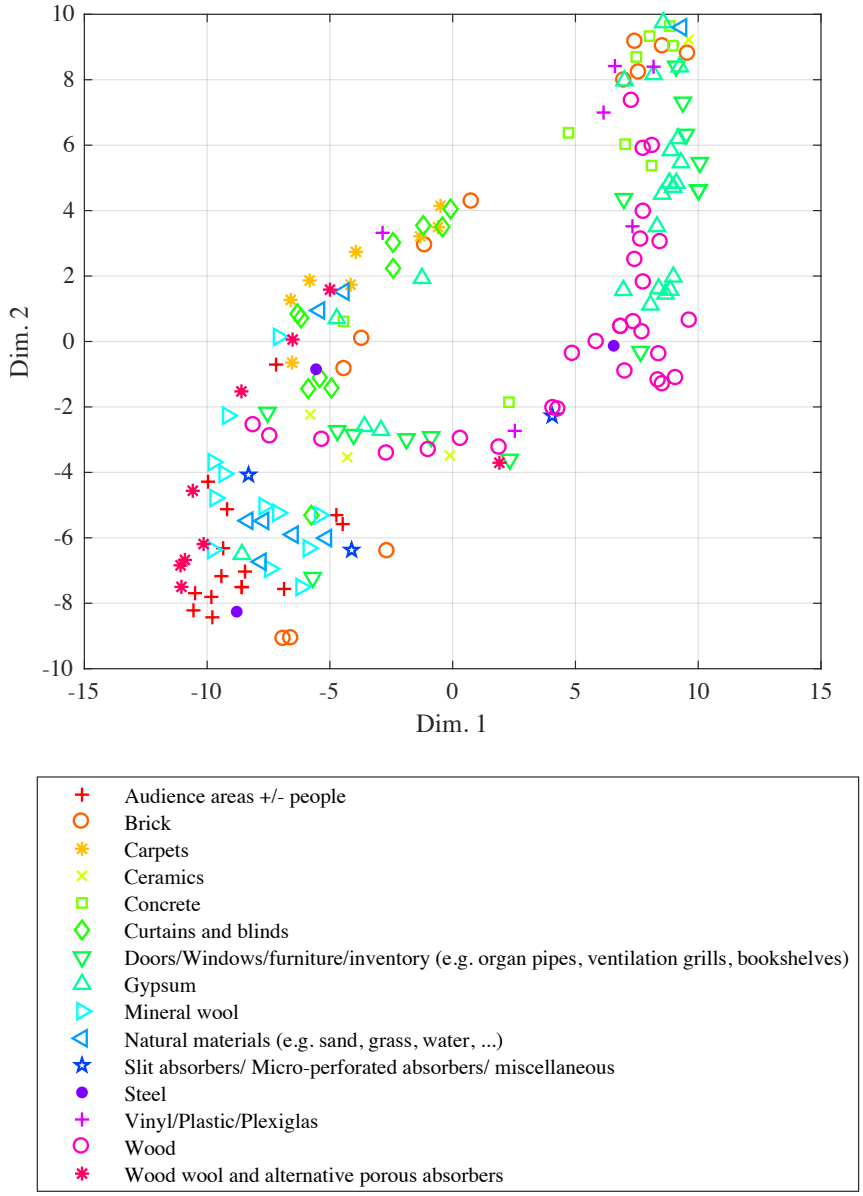


Figure 8.4: t-SNE data visualisation of different materials based on their frequency-dependent absorptions, grouped by the type-labels provided with the database.

is therefore restated to detecting the presence of materials in groups, that have the same or similar absorptions characteristics. The effect of materials on sound which is within the scope of this work is the frequency-dependent absorption of the sound's energy. Grouping materials together that have similar absorption properties therefore still provides a solution to the task. A grouping scheme needs to be therefore implemented, which takes as inputs the collection of absorption coefficients of the 163 materials and returns a smaller number of absorption values. Each value will represent of a subset of the 163 materials.

Firstly, it is naively considered grouping together materials of the same type. The result of this grouping is shown using t-SNE visualisation [123] in Figure 8.4. It can be seen that materials of the same type do not form well-defined groups. The use of unsupervised learning for the grouping of materials into a small number of groups is therefore explored as the next Section.

8.3.2 Clustering and choosing number of clusters

In the previous Section the need to create groups of materials, characterised by similar frequency-dependent absorptions was highlighted. k-means [124] is used for creating groups of materials as clusters. k-means does not identify the number of groups in the data but provides a grouping solution, given the desired number of groups. In Section 3.3, the Davies-Bouldin criterion and the VRC are described as tools that are used to evaluate the number of clusters, given a set of unlabelled data. They are used here to identify the suitable number of clusters to use in this case.

The absorption coefficients of 163 materials in the 8 1-octave bands are clustered by k-means algorithm for a range of a number of clusters between 2–80. The results are shown in Figure 8.5. The different formulations of the two criteria give competing results. The optimal values for each are at the extremes, however if the trend of the two lines is ignored, the 10 cluster solution gives a *knee* in the curve of both criteria [84]. $K_{\text{opt}} = 10$ is chosen therefore as the number of clusters and used to separate the 163 materials in 10 groups.

The result of clustering the data in 10 groups is shown in Figure 8.6. The mean of

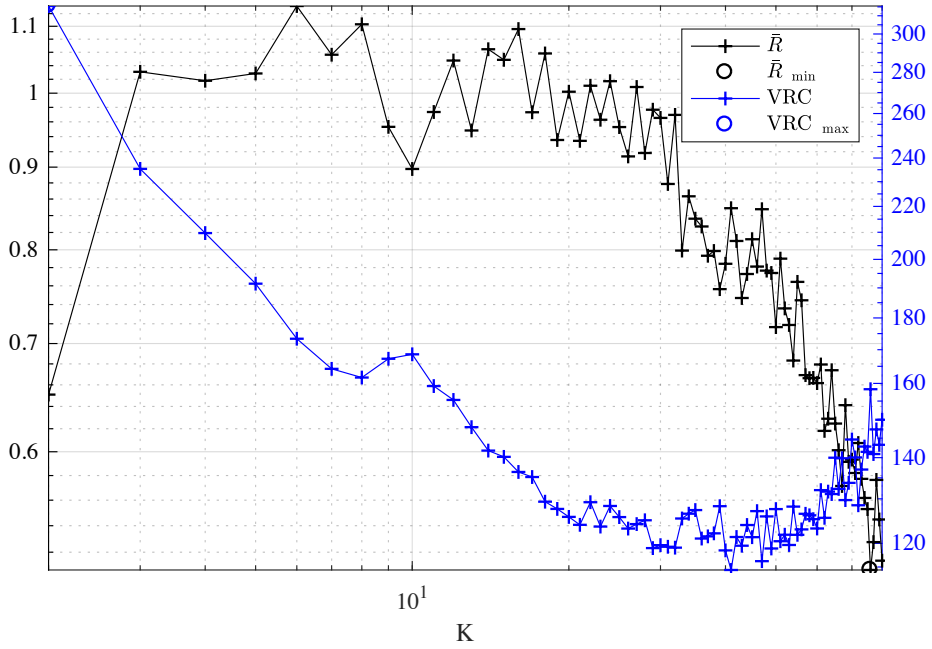


Figure 8.5: Evaluating the quality of k-means clusters for materials, based on their frequency-dependent absorptions. The algorithm clusters 163 materials for a varying number of clusters in the range 2–80.

each group will represent a material frequency-dependent absorption to be detected by the DNN discussed in Section 8.2.3. This clustering forms a segmentation of the space in the original 8-dimensional space of the absorptions. The DNN will therefore be trained to indicate the cluster whose mean is closest to the absorption of the material in the environment. The means of the clusters to be detected by the DNN are shown in Figure 8.7. To link this data collection part back to the detection-model notation, the results of the k-means clustering is used for the construction of matrix $\mathbf{A}_{\text{tot}} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{\Theta_{\text{tot}}}]^T$. The total number of material absorptions to be detected will be given as $\Theta_{\text{tot}} = K_{\text{opt}} = 10$, equal to the number of clusters chosen from k-means.

With the detector DNN for materials defined and the source for the training data identified, the next Section will present how these detections are used in order to estimate the parameters of early reflections. The work of Chapter 7 is enhanced with the use of knowledge of the materials present in the room, to provide estimates about the ToAs and scales of reflections and information about their interaction with the present materials.

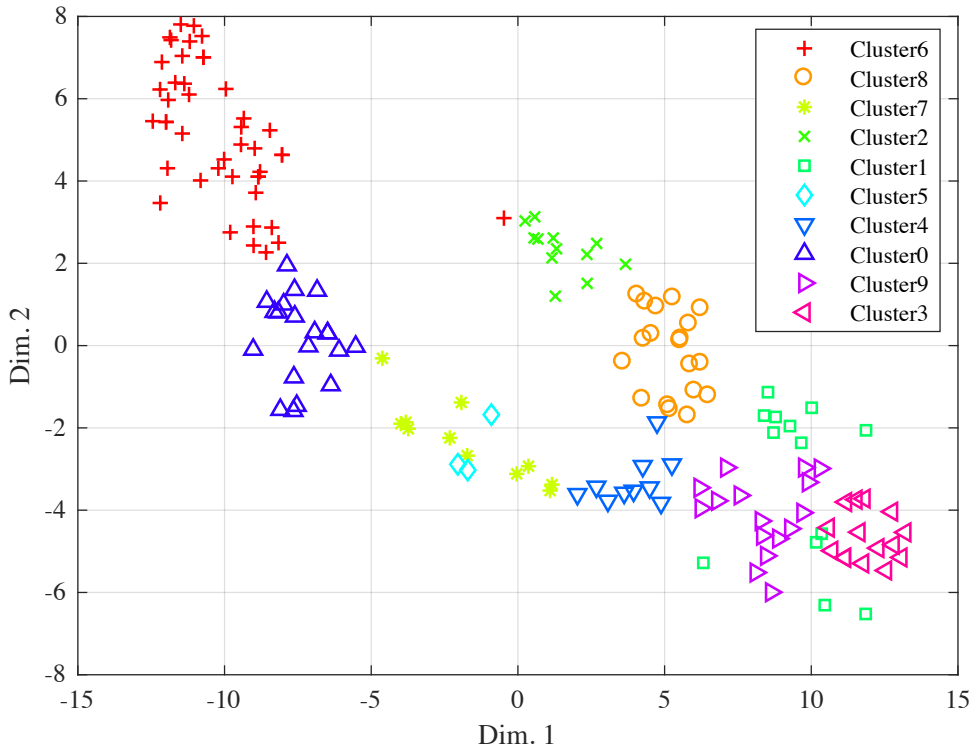


Figure 8.6: t-SNE data visualisation of 163 materials based on their frequency-dependent absorption values, clustered by k-means into 10 clusters. The number of clusters was chosen by examining the value of the Davies-Bouldin criterion and the VRC for clustering solutions with clusters in the range 2–80.

8.4 Estimating sound and material interaction

The previous Sections have proposed a novel way for estimating the frequency-dependent absorptions of the materials present in an environment, using the taps of an AIR. The aim is to better model the early reflection. Chapter 7 provided a method for estimating the ToAs and scales of reflections in an AIR and the excitation used to measure it. The estimation of these parameters and the modelling of reflections can be improved by considering the effect of materials on the reflections. Integrating the estimation method of Chapter 7 with the information about the materials is the topic of this Section, which proposes a novel method for modelling early reflections in a material-aware manner.

The diagram of Figure 8.8 illustrates the components of the reflection parameter estimation method proposed in this Chapter. The process uses a given AIR and performs

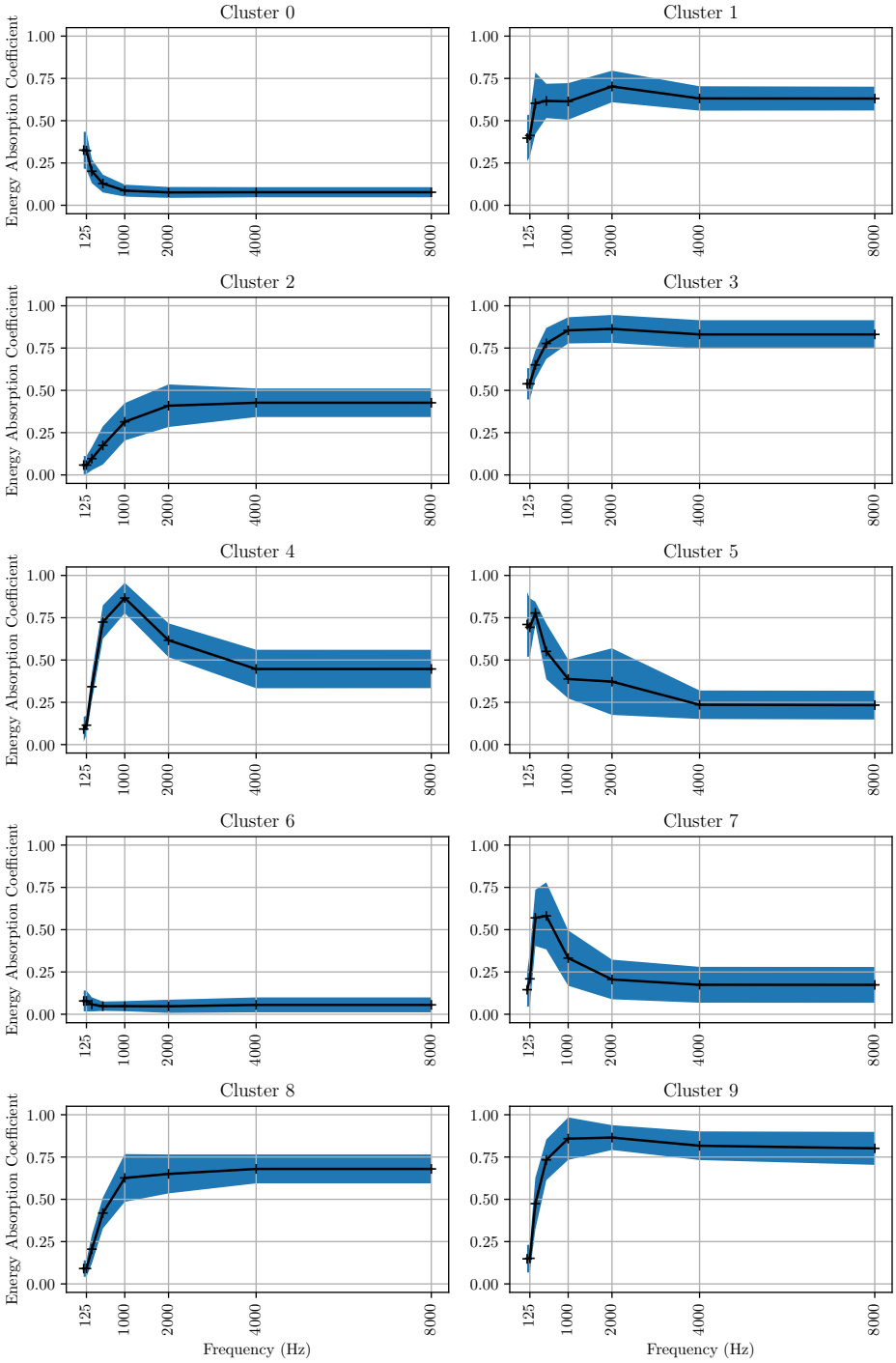


Figure 8.7: Mean energy-absorption coefficients of k-means derived clusters. 10 clusters are selected to describe the training data consisting of 163 materials, based on their frequency dependent absorptions. The presence of these clusters in the acoustic environment will be detected by the proposed DNNs. The solid lines show the means of the clusters and the shading shows the corresponding standard deviation.

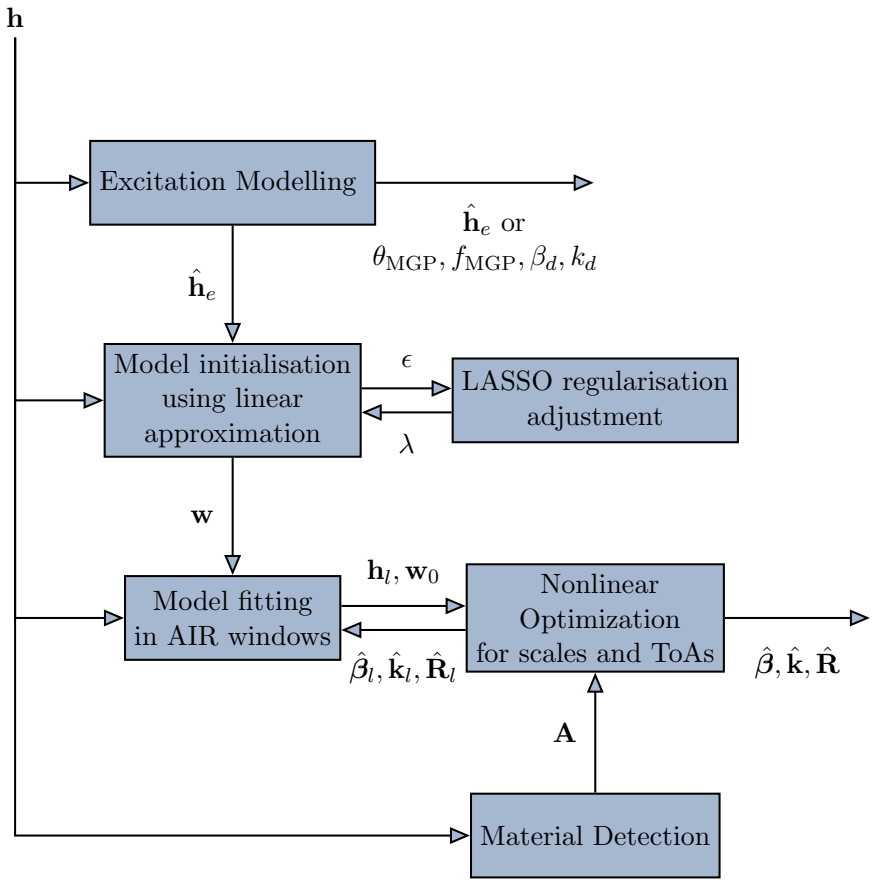


Figure 8.8: Diagram of proposed method for modelling the early part of AIRs. The model characterises individual reflections based on their ToAs and scales and their interaction with the materials present in the environment.

the following steps:

1. Estimate the excitation used to measure the AIR as $\hat{h}_e(n)$, as discussed in Section 7.3.
2. Initialise the ToAs and scales of reflections as $\boldsymbol{k}_R$ and $\boldsymbol{\beta}_R$ respectively, as described in Section 7.4.1.
3. Estimate the frequency-dependent absorptions of the materials present in the environment and construct the corresponding matrix of absorptions $\mathbf{A}$, using the models proposed in Section 8.2.3.
4. Optimise the ToAs $\hat{\boldsymbol{k}}$, scales $\hat{\boldsymbol{\beta}}$ and material reflection counts $\hat{\mathbf{R}}$ for each reflection.

Step 4 is described in the following Section. The consideration of frequency-dependent absorption changes this part of parameter estimation from Chapter 7, which was originally discussed in Section 7.5. The first two steps remain unchanged and the reader can refer to the appropriate Sections for more information.

8.4.1 Optimising reflection parameters

This optimisation discussed in this section estimates the delays $\hat{\boldsymbol{k}}$, scales $\hat{\boldsymbol{\beta}}$ and material reflection counts $\hat{\mathbf{R}}$ of early reflections. Optimising the parameters is split into two parts. The first part optimises the material reflection-counts and the second part fine-tunes the delays and scales of reflections. This Section first explains in detail the process of estimating $\hat{\mathbf{R}}$ and then the remaining process is given.

The interaction between reflections and the materials in the room is summarised by matrix $\mathbf{R}$. The element at row i and column θ of this matrix will hold the number of times the reflection i was reflected off surfaces of material θ, with absorption filter $\phi_\theta(n)$. The elements of the matrix are therefore positive integers and the sum of each of the rows will give the order of the corresponding reflection Ξ_i.

The estimation of $\hat{\mathbf{R}}$ is based on the use of the absorption filters $\hat{\mathbf{\Phi}}$, to reconstruct the spectrum of windows of the AIR. Reflection counts are found, which describe the number of times each filter in $\hat{\mathbf{\Phi}}$ is used to reconstruct the spectrum, using (8.5). For an AIR window $h_l(n)$, its DFT is denoted by $H_l(u)$. The presence of a single reflection in the window is assumed, of an unknown order. To find the number of times the reflection was reflected off each material, a minimisation problem is posed. The L_1 distance is to be minimised, between a reconstruction of the spectrum and the spectrum of the window itself, by adjusting the reflection-count per material. The objective is to find

$$\begin{aligned}(\hat{\mathbf{r}}_l, \beta_0) &= \underset{\hat{\mathbf{r}}_l, \beta_0}{\operatorname{argmin}} \left(\left| \hat{H}_l(u) \right| - \beta_0 \left| H_e(u) \right| \left| \prod_{\theta=1}^{\hat{\Theta}} \Phi_\theta^{r_{\theta,l}}(u) \right| \right) \\ &= \underset{\hat{\mathbf{r}}_l, \beta_0}{\operatorname{argmin}} \left(\left| H_l(u) \right| - \left| \hat{H}_l(u) \right| \right).\end{aligned} \tag{8.11}$$

The problem is not only non-linear but it is also integer constrained with respect to

$\hat{\mathbf{r}}_l$. The non-linearity is addressed using the properties of the log operators which turns the problem into a linear form. Applying the log operator on the reconstruction of the spectrum $\left|\hat{H}_l(u)\right|$ turns it into

$$\log\left|\hat{H}_l(u)\right| = \log\beta_0 + \log|H_e(u)| + \sum_{\theta=1}^{\hat{\Theta}} \hat{r}_{\theta,l}\log|\Phi_\theta(u)| \tag{8.12}$$

and therefore (8.11) turns into

$$(\hat{\mathbf{r}}_l, \beta_0) = \underset{\hat{\mathbf{r}}_l,\beta_0}{\operatorname{argmin}} \left(\log|H_l(u)| - \log\beta_0 + \log|H_e(u)| + \sum_{\theta=1}^{\hat{\Theta}} \hat{r}_{\theta,l}\log|\Phi_\theta(u)| \right), \tag{8.13}$$

which involves now linear terms for the integers in $\hat{\mathbf{r}}_l$ and solving for $\log\beta_0$ trivially provides β_0. The term β_0 refers to the scaling coefficient of the spectrum, which will account for the remaining mismatch. The remaining mismatch is attributed to scattering and other factors, as introduced in Section 8.1.1.

This linear problem is still constrained for integrality on $\hat{\mathbf{r}}_l$. These problems are typically addressed using Mixed Integer Programming (MIP) and for its solution, the Branch-and-Cut solver [162] is used[3]. The formulation of the optimisation problem has to be modified from the one given in (8.13). The MIP solver cannot jointly optimise the response at each frequency bin u but minimises a single scalar objective. Summing over u in (8.13) collapses all the terms together as the same terms appear across all bands, losing the meaning of spectral shape. The minimisation of (8.13) is therefore approximated using a strategy which involves maximising $\log\beta_0$, with a number of constraints equal to the number of bands. These constraints specifying the gap between the desired response $\log|H_l(u)|$ at each bin and the reconstruction through the remaining terms. This way, the objective of the optimisation is to find estimates of β_0 and $\hat{\mathbf{r}}_l$, which best fit the desired response by minimising the energy loss due to factors not related to material absorption. The proposed strategy is therefore to perform

$$(\hat{\mathbf{r}}, \beta_0) = \underset{\hat{\mathbf{r}},\beta_0}{\operatorname{argmax}}\,\beta_0 \tag{8.14}$$

[3]As discussed and implemented in an open source format `https://projects.coin-or.org/Cbc`.

s.t. the constraints

$$H_l(u) - \hat{H}_l(u) <= \epsilon_{\max} \; \forall \; u \tag{8.15}$$

$$- H_l(u) + \hat{H}_l(u) <= \epsilon_{\max} \; \forall \; u, \tag{8.16}$$

where $\epsilon_{\max}$ is the defined maximum deviation between the estimated and the observed response of the window. For the optimisation, the value of $\epsilon_{\max}$ is initially set to a small value and it is increased until the optimisation is feasible within the constraints. This way, the aim is to minimise the energy loss not explained by the surface interaction and ensure an accurate spectral representation.

The previous optimisation step described how the spectrum of an AIR window can be explained by the absorptions of the materials present in an environment. This provides estimates for the number of times $r_{\theta,i}$ a reflection i in the window is reflected off material θ and initial values for the reflection scale β_0. The second and final step of the optimisation optimises the ToAs and scales of the reflections. The initial estimation of the ToAs of reflections is made using LASSO, from Section 7.4.1. These initial estimates are optimised, similarly to Section 7.5. The only addition is the material filter in the reconstruction of the window given by (7.14), as presented in (8.4). The interior-point [153] optimisation solves for the delay $\hat{k}_l$ and scale $\hat{\beta}_l$ of a reflection, by minimising the MSE between the AIR window $h_l(n)$ and its reconstruction, now in the time domain. Repeating the process for all windows results in an estimate for the ToAs and scales for the $\hat{D}$ reflections. The AIR taps can be reconstructed from the optimised parameters using

$$\hat{h}(n) = \sum_{i=1}^{\hat{D}} \hat{\beta}_i \hat{h}_e(n) * \frac{\sin\left[\pi(n-\hat{k}_i)\right]}{\pi(n-\hat{k}_i)} \; \mathfrak{F}^{-1}\left(\prod_{\theta=0}^{\Theta_{\text{tot}}-1} \Phi_\theta^{r_{\theta,i}}(u) \right). \tag{8.17}$$

The filtering with $\Phi_\theta(u)$ is done by scaling the energy of windows in the frequency domain using the energy absorption values of Figure 8.7. Since the requirements are specified only in the magnitude-frequency domain, the filtering is performed in the same domain, as described below, which avoids the explicit design of filters. The filtering process for a reflection i uses the following information:

- The vector $\mathbf{r}_i$, containing the Θ integer material reflection-counts.
- The specification of the energy-absorption coefficients for each of the material clusters in vectors $\mathbf{a}_\theta \; \forall \; \theta \in \{1, 2, \ldots, \Theta\}$, for octave bands centred at $\mathbf{f}_{\text{oct}} = \{63, 125, 250, 500, 1000, 2000, 4000, 8000\}$ Hz.
- The excitation estimate $\hat{h}_e(n)$ with N_d samples.
- The sampling frequency F_s

The filtering process for a reflection i follows the steps:

1. Take the DFT $H_i(u) = \mathfrak{F}\{h_e(n)\}$.
2. For each bin u, referring to frequency $f_u = u\frac{N_d}{F_s/2}$, find its 2 closest values in $\mathbf{f}_{\text{oct}}$ and identify them as f_{-u} and f_{+u} such that $f_{-u} < f_u$ and $f_{+u} > f_u$.
3. Iterating over $\theta \in \{1, \ldots, \Theta\}$ material types present and for each bin u
   (a) Estimate the energy absorption coefficient due to material type θ at the specific bin via the interpolation
   $$\hat{a} = \left(\frac{f_u - f_{-u}}{f_{+u} - f_{-u}}\right)\mathbf{a}_\theta[f_{-u}] + \left(\frac{f_{+u} - f_u}{f_{+u} - f_{-u}}\right)\mathbf{a}_\theta[f_{+u}]. \tag{8.18}$$
   (b) Update $H_i(u) = (1 - \hat{a})^{\frac{r_{i,\theta}}{2}} H_i(u)$.
4. Reconstruct reflection i as
   $$\hat{\beta}_i h_i(n) * \frac{\sin\left[\pi(n - \hat{k}_i)\right]}{\pi(n - \hat{k}_i)}. \tag{8.19}$$

This process conveys the effect of sound energy absorption without affecting the phase of the excitation signal. Altering the phase shifts the ToA of the reflection, which causes a significant mismatch between the LASSO estimates and the result.

This discussion concludes this Section, which has discussed how estimates of the frequency-dependent absorptions of materials present in the environment can be used to estimate parameters of the early reflections. The estimation starts from a given AIR and

	Total	Train	Valid.	Test
Percentage	100%	85%	7.5%	7.5%
AIRs	70,500	59,925	5,287	5,288

Table 8.1: Partitioning of AIRs into sets for the training and evaluation of DNNs. AIRs are simulated in shoe-box rooms, using known material frequency dependent absorptions.

the parameters form a parametric model, able to reconstruct taps of the original AIR. The next Sections of this Chapter will investigate the performance of the proposed material detection DNN and the descriptive accuracy of the parameters estimated to describe the early reflections.

8.5 Material type detection experiments

This Chapter proposed a novel method for estimating the frequency-dependent absorptions in an acoustic environment, which is based on detecting the presence of materials using their sound absorption properties. This Section describes the experiments performed to evaluate the performance of the material detection DNNs of Section 8.2.3. The networks accept an AIR as the input and output the estimated posterior probability of presence of the 10 material clusters shown in Figure 8.7.

Simulated AIRs are used as the training examples for the networks, which have been generated using [155], with perfect knowledge of the materials present in the environments. The training AIRs are generated by simulating 141 three-dimensional *shoe-box* rooms with walls of known frequency-dependent absorptions, using [155]. The size of the simulated rooms is random-uniformly chosen between [2.5, 2.5, 2.5] and [7.0, 7.0, 2.6] m. For each one of the 6 walls of the room, the frequency-dependent absorptions are chosen from one of the 163 materials in the list described in Section 8.3. Each room is populated with 10 sources and 5 receivers at random locations. Collecting the AIR between each source and receiver pair results in 70,500 AIRs. Each one serves as an in individual training example. The sampling rate used is 16 kHz.

The 70,500 generated AIRs are split into 3 sets, the training, test and validation set. The data partitioning is shown in Table 8.1. The aim is for each one of the sets to be

Material Cluster	Average Proportion	Train Positives	Validations Positives	Test Positives	Cluster Size
0	41.8%	34,850	3,070	3,080	20
1	23.7%	19,719	1,743	1,738	13
2	21.4%	17,750	1,577	1,573	12
3	21.3%	17,671	1,565	1,564	15
4	25.9%	21,581	1,909	1,910	10
5	6.9%	5,686	507	507	3
6	63.9%	53,308	4,696	4,696	44
7	27.2%	22,696	2,003	2,001	11
8	34.0%	28,358	2,497	2,495	19
9	34.0%	28,305	2,498	2,497	16
0–9	100%	59,925	5,287	5,288	163

Table 8.2: Positive samples per material cluster per partition in the set of AIRs, which are simulated in shoe-box rooms, using known material frequency dependent absorptions.

a representative subset of the overall population. Stratified partitioning is therefore used between the three sets [163], which preserves the class ratios. In these experiments, the number of material absorptions to be detected is $\Theta = 10$ and the resulting distribution of positive samples for each, across partitions, is shown in Table 8.2.

The detector models are trained using the Adam [59] optimizer with a binary cross-entropy loss [63]. The configuration of the optimizer is set up as proposed in the original paper. The batch size is set to 128 AIRs of duration 200 ms, which are split into frames of 3 ms with 1.5 ms overlap. Training is done using Tensorflow [164] on an NVIDIA GeForce GTX 980 GPU. Overfitting is prevented by early stopping [63] which stops the training of the model 10 epochs after the training loss stopped improving or 15 epochs after the validation loss stopped improving. The final model is selected at the epoch with the minimum validation loss. To improve generalisation, the models use a dropout mechanism for the FF layers, as discussed in [46]. The dropout rates are shown in Figure 8.1. Table 8.2 has shown that the distribution of positives samples across labels is not even and the positive and negative samples within labels are not balanced. To convey this information to the training process and avoid skewing of the decisions of the model, the contribution to the cross-entropy loss of AIR $h_m(n)$ is weighted by w_m, as proposed in [165] and given

	Overall	Material Type θ									
		0	1	2	3	4	5	6	7	8	9
F1 (%)	98	99	97	97	97	98	98	100	98	98	98
Precision (%)	98	98	97	97	99	98	99	100	99	98	98
Recall (%)	98	99	97	96	95	98	98	100	97	98	99

Table 8.3: FF-RNN material detector test performance on 5,288 test AIRs.

	Overall	Material Type θ									
		0	1	2	3	4	5	6	7	8	9
F1 (%)	98	98	97	96	97	98	98	100	97	98	98
Precision (%)	98	98	95	97	97	98	98	100	99	99	97
Recall (%)	98	99	98	96	98	99	98	100	96	98	98

Table 8.4: CNN-RNN material detector test performance on 5,288 test AIRs.

by

$$w_m = \prod_{\theta=1}^{\Theta_{\text{tot}}} \frac{M}{\Theta_{\text{tot}} \sum_{\tilde{m}=1}^{M} \{y_{m,\theta} y_{\tilde{m},\theta} + (1 - y_{m,\theta})(1 - y_{\tilde{m},\theta})\}}, \tag{8.20}$$

where M the total number of training examples.

The detection performance of the model is measured using precision and recall [166] and their harmonic mean, the F_1 score [167]. The definition of the measures is respectively

$$P = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}, \tag{8.21}$$

$$R = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8.22}$$

and

$$F_1 = \frac{2PR}{P + R}. \tag{8.23}$$

The results of evaluating the models based on the above measures are given in Tables 8.3 and 8.4, for the FF-RNN and CNN-RNN respectively. The overall scores show the models being equally accurate when comparing their overall F_1 scores. The same applies to individual classes with a small number of exemptions. The differences are never greater than 1%. In terms of individual material types, unsurprisingly the best results are obtained for $\theta = 6$, which is the biggest cluster, containing 63.9% of the training examples. The evaluation was performed for a threshold of 0.5 across all outputs.

Given the similar test accuracy of the two models, the CNN-RNN is used in the following Sections as the material type detector. This model accepts an input AIR and estimates the frequency-dependent absorptions of the materials present in the environment by detecting the presence of material clusters. The absorptions will be provided to the AIR modelling algorithm, which will use them to associate identified reflections to materials. The following experiments evaluate the performance of this modelling method.

8.6 Reflection modelling experiments

The aim of detecting materials present in the room is to estimate the frequency-dependent absorptions in the environment. This information is used by the proposed method for estimating early reflection parameters in this Chapter. This Section describes experiments that evaluate the performance of modelling early reflections using the estimated parameters. The experiments use simulated data and also measured AIRs in real rooms. This will demonstrate how the performance varies between the two and how non-idealities in measured responses affects it.

The setting of hyper-parameters of the modelling, such as the regularisation adjustment for LASSO, have remained unchanged from the experiments of Chapter 7. The thresholds of the outputs of the material detector are set to 0.5. The sampling rate is 16 kHz throughout and the first 24 ms after the arrival of the direct sound of each AIR is modelled unless stated otherwise.

8.6.1 Modelling of simulated AIRs

To evaluate the proposed modelling of early reflections 400 AIRs are randomly chosen from the 5,287 AIRs simulated for the detector network testing, in Table 8.1. For the evaluation measures, the Normalized Projection Misalignment (NPM) [168] and the Itakura distance [169] are used. They evaluate the fit in the time and spectral domain, both of which are

aspects considered in this work. The NPM is defined as [168]

$$\xi_{\text{NPM}} = 1 - \left(\frac{\mathbf{h}^T \hat{\mathbf{h}}}{||\mathbf{h}||\ ||\hat{\mathbf{h}}||} \right)^2 \tag{8.24}$$

and the Itakura distance is defined as

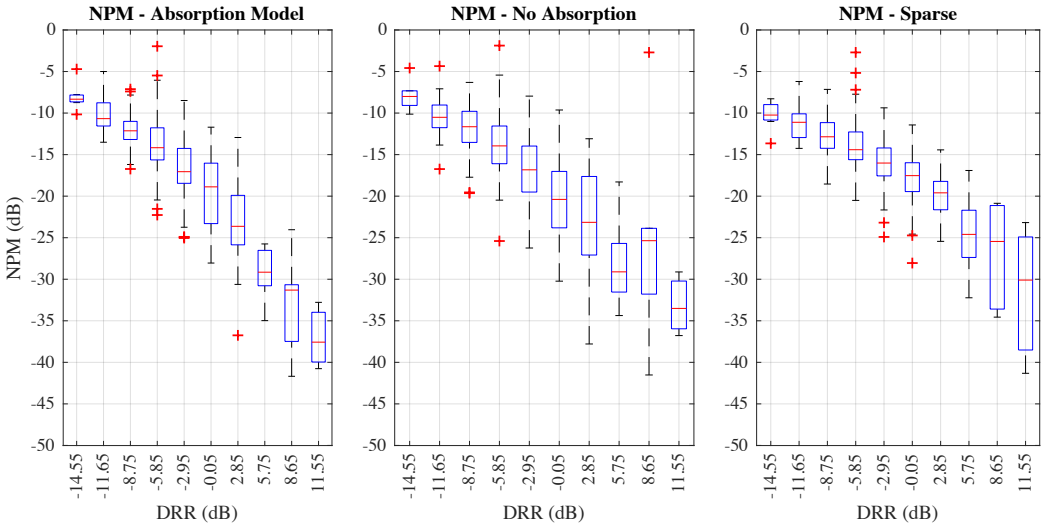
$$d_{\text{I}} = \log \frac{1}{N/2} \sum_{u=0}^{N/2} \frac{\mathbf{H}_p(u)}{\hat{\mathbf{H}}_p(u)} - \frac{1}{N/2} \sum_{u=0}^{N/2} \log \frac{\mathbf{H}_p(u)}{\hat{\mathbf{H}}_p(u)}, \tag{8.25}$$

where $\mathbf{H}_p$ and $\hat{\mathbf{H}}_p$ are the power spectra of $\mathbf{h}$ and $\hat{\mathbf{h}}$ respectively. $\hat{\mathbf{h}}$ denotes the estimate of $\mathbf{h}$.

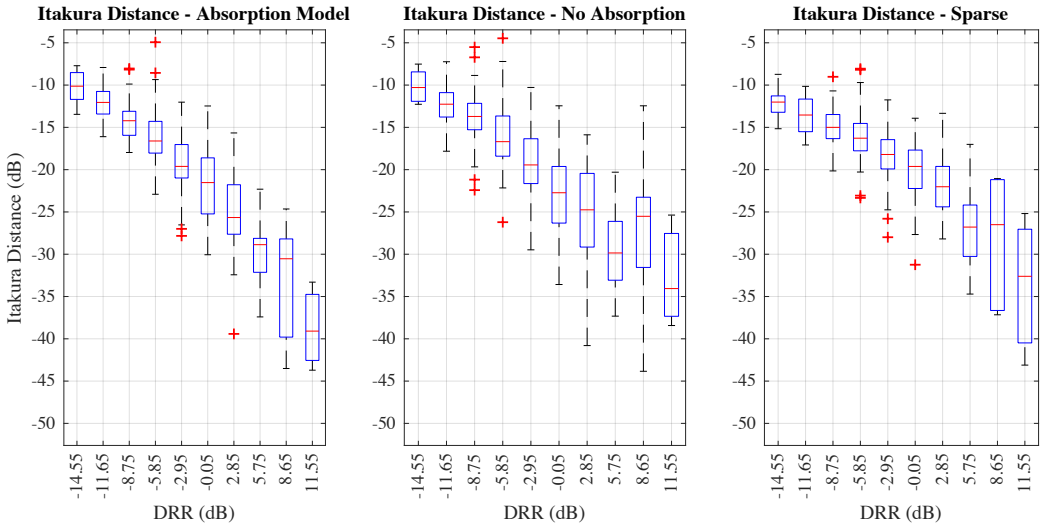
In this experiment, the performance of the following 3 modelling approaches for early reflections is compared.

1. The method proposed in this Chapter.
2. The method proposed in Chapter 7.
3. A purely sparsity driven model, with no link to reverberation. The inclusion of this model to the comparison asks the question, "What is the benefit of modelling AIRs using the general concept of reflections?". The model simply preserves the $\frac{N_{\text{sparse}}}{2}$ highest energy samples of the AIR and represents them with their sample index and their amplitude. The rest are set to 0. The model therefore uses a total of N_{sparse} parameters. The aim is to show that the proposed model can describe the AIR at least as well as the sparse representation, using the same number of coefficients, but at the same time managing to provide a meaningful representation with regards to the environment. Therefore, N_{sparse} is fixed for each AIR to the same number of parameters estimated by the proposed method to model the reflections.

The results of modelling 400 simulated AIRs using the 3 models involved in this experiment are shown in Figure 8.9 and summarised in Table 8.5. The Figure shows that both NPM and Itakura distance are negatively correlated to the DRR. The negative correlation is the highest for the proposed method, with a Pearson correlation $\rho = -0.82$ for both. The proposed model outperforms the rest for most positive DRR environments



(a) NPM for 400 simulated responses grouped by DRR.



(b) Itakura distance for 400 simulated responses grouped by DRR.

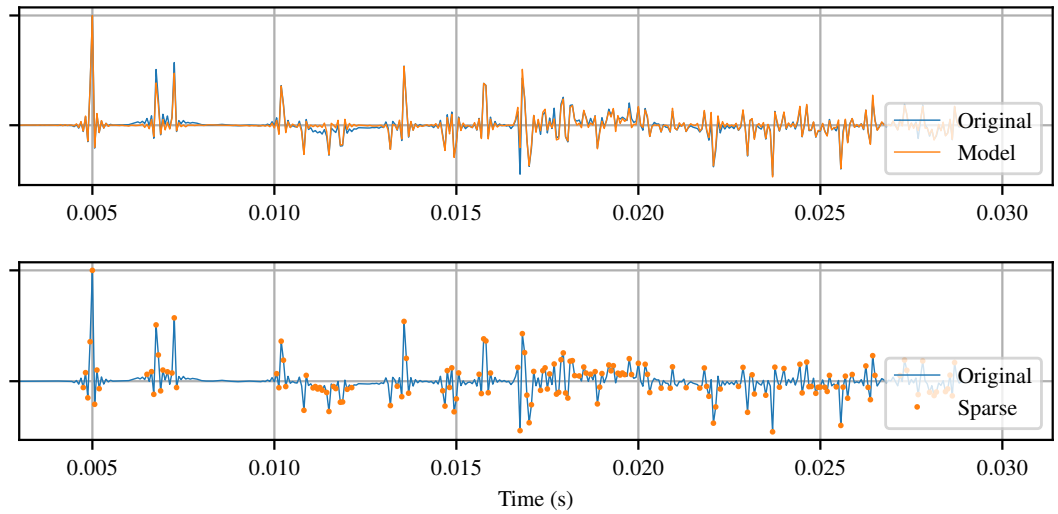
Figure 8.9: Performance of modelling early reflections using the proposed model with and without frequency-dependent absorptions and comparison with a simple sparse representation.

	Proposed Absorption		Proposed No Absorption		Sparsity driven	
Median DRR (dB)	NPM	Itak. Dist.	NPM	Itak. Dist.	NPM	Itak. Dist.
-14.55	-8.33	-10.13	-8.00	-10.29	**-10.24**	**-12.01**
-11.64	-10.67	-12.05	-10.51	**-12.26**	-11.11	**-13.54**
-8.75	-12.14	-14.22	-11.65	-13.71	**-12.86**	**-15.00**
-5.84	-14.17	-16.61	-13.95	**-16.69**	**-14.41**	-16.28
-2.95	**-17.06**	**-19.62**	-16.83	-19.44	-16.02	-18.20
-0.05	-18.88	-21.53	**-20.39**	**-22.73**	-17.53	-19.62
2.85	**-23.63**	**-25.66**	-23.15	-24.75	-19.60	-22.02
5.75	**-29.15**	-28.87	-29.11	**-29.85**	-24.60	-26.79
8.65	**-31.30**	**-30.53**	-25.36	-25.52	-25.45	-26.51
11.55	**-37.57**	**-39.10**	-33.51	-34.06	-30.10	-32.61
DRR Corr. (ρ)	**-0.82**	**-0.82**	-0.75	-0.75	-0.76	-0.73

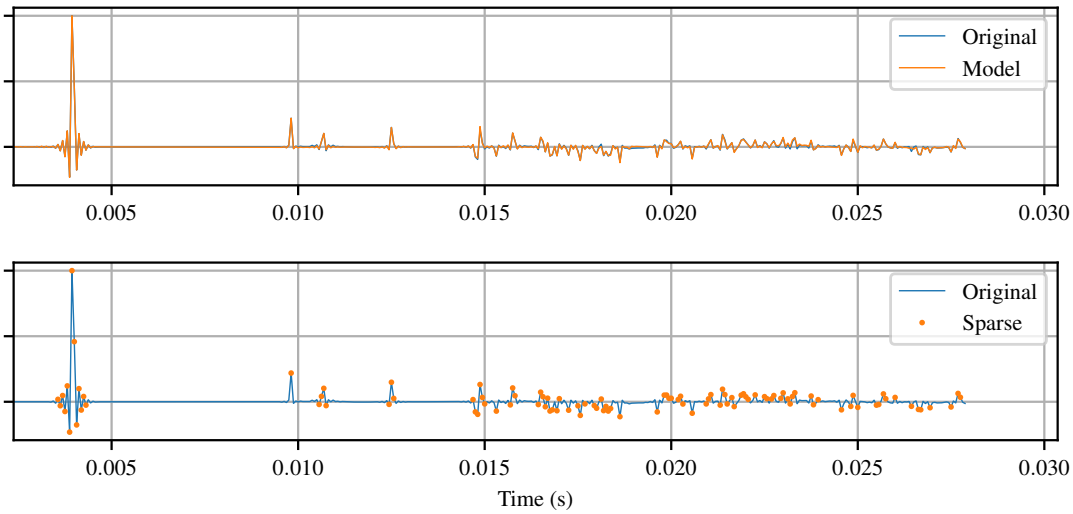
Table 8.5: Performance of proposed model with and without frequency dependent absorptions for simulated AIRs and comparison with simple sparse representation. Values show the mean NPM and Itakura distance for groups of AIRs with the indicated median value for the DRR.

and the sparsity driven model outperforms the rest for most negative DRR environments.

To investigate the link between the DRR and the modelling performance, Figure 8.10 shows two visualisations of the result of the modelling. One example with a negative and one example with a positive DRR, with the results compared between the sparsity driven and the proposed model. The results show that for the high DRR case, the proposed model identifies the small number of reflections present in the early part and therefore uses a small number of parameters to accurately model them. The opposite is true for the case of low DRR, where the proposed method has to disambiguate many reflections with high overlap, leading to large parameter counts and reduced modelling accuracy. The sparsity driven model simply captures the energy at its peaks without the need to estimate any parameters. This provides an accurate temporal representation but without any semantic meaning.



(a) Modelling an AIR with negative DRR (-9.22 dB) using the proposed method (top) and using a simple sparse representation (bottom).

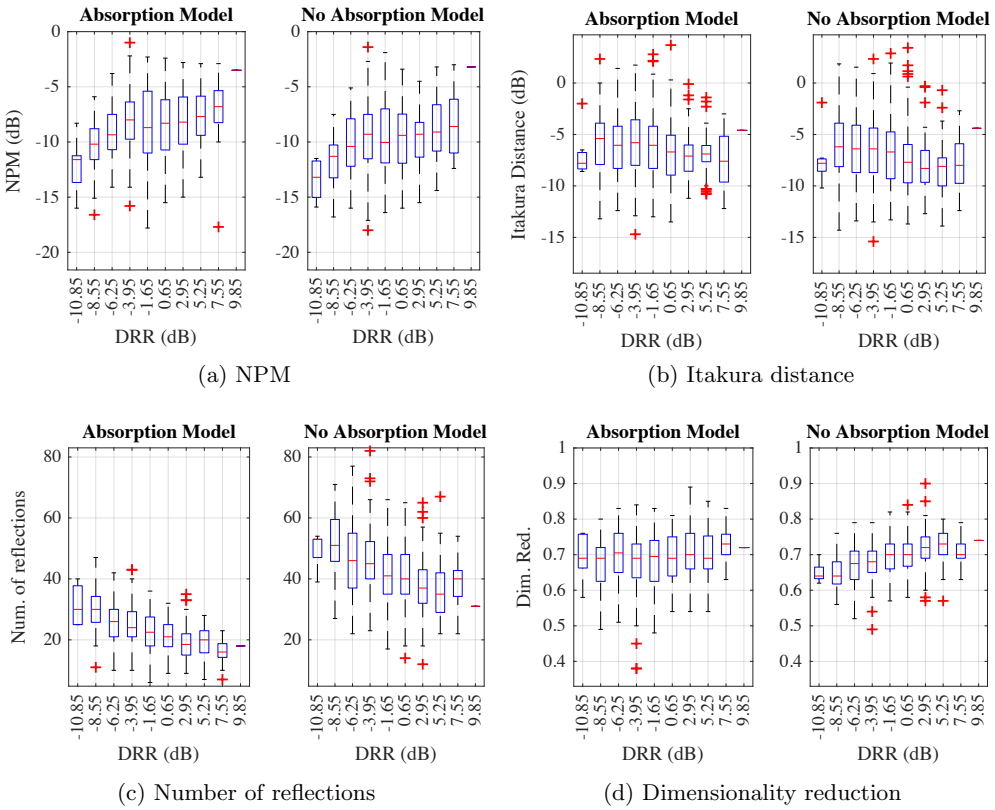


(b) Modelling an AIR with positive DRR (2.88 dB) using the proposed method (top) and using a simple sparse representation (bottom).

Figure 8.10: Modelling AIRs with negative and positive DRR using the proposed method and a sparse representation.

8.6.2 Modelling of measured AIRs

The previous Section investigated the performance of modelling early reflections when dealing with simulated responses. This Section investigates the modelling of measured AIRs. All AIRs part of the ACE challenge database [11] are used in this experiment, summarised in Table 5.2. The database provides 700 AIRs, 100 from each of the 7 rooms



(a) NPM (b) Itakura distance

(c) Number of reflections (d) Dimensionality reduction

Figure 8.11: NPM, Itakura distance, number of reflections detected and dimensionality reduction for modelling of the early part of measured AIRs of the ACE database. The results show the performance of modelling early reflections using the proposed model with and without frequency-dependent absorptions

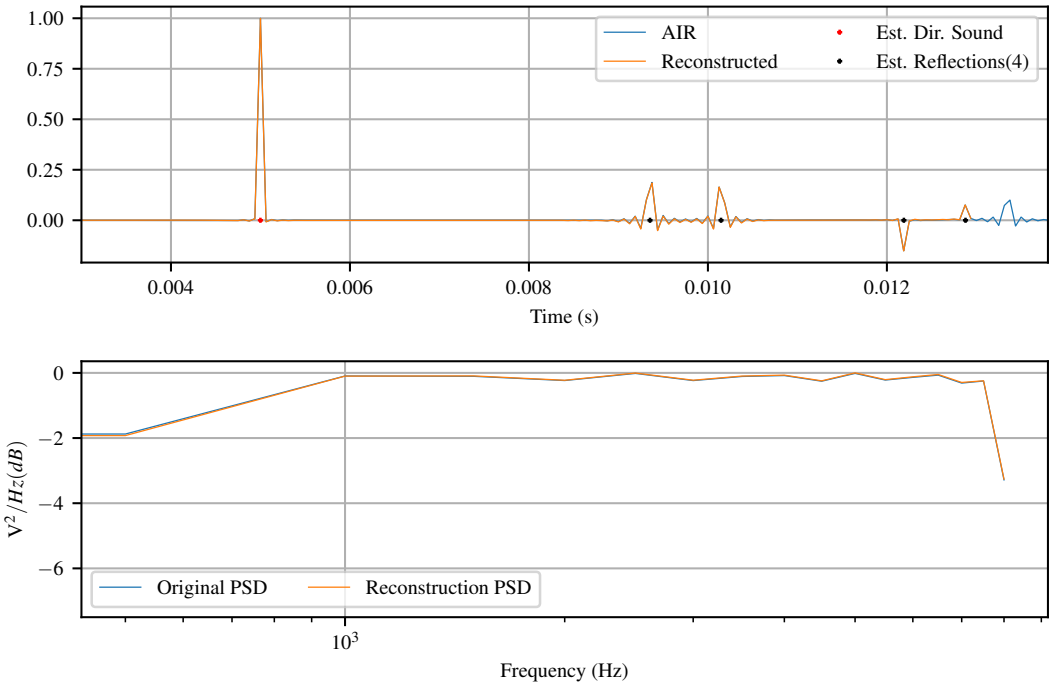
involved in the measurements.

The NPM, Itakura distance, the number of reflections and the dimensionality reduction offered by the modelling are shown in Figure 8.11. The Figure shows a side-by-side comparison of the result of considering frequency-dependent absorptions and of not considering them. The accuracy of modelling reflections using either of the two methods provides similar NPM values and Itakura distances, with the median differences being 1 dB and 0.5 dB respectively, in favour of the model which does not model absorptions. This however comes at the expense of using approximately double the number of reflections to represent the AIR. The proposed model used on average 55% of the number of reflections used by the alternative to model the same AIRs. The results therefore show that consid-

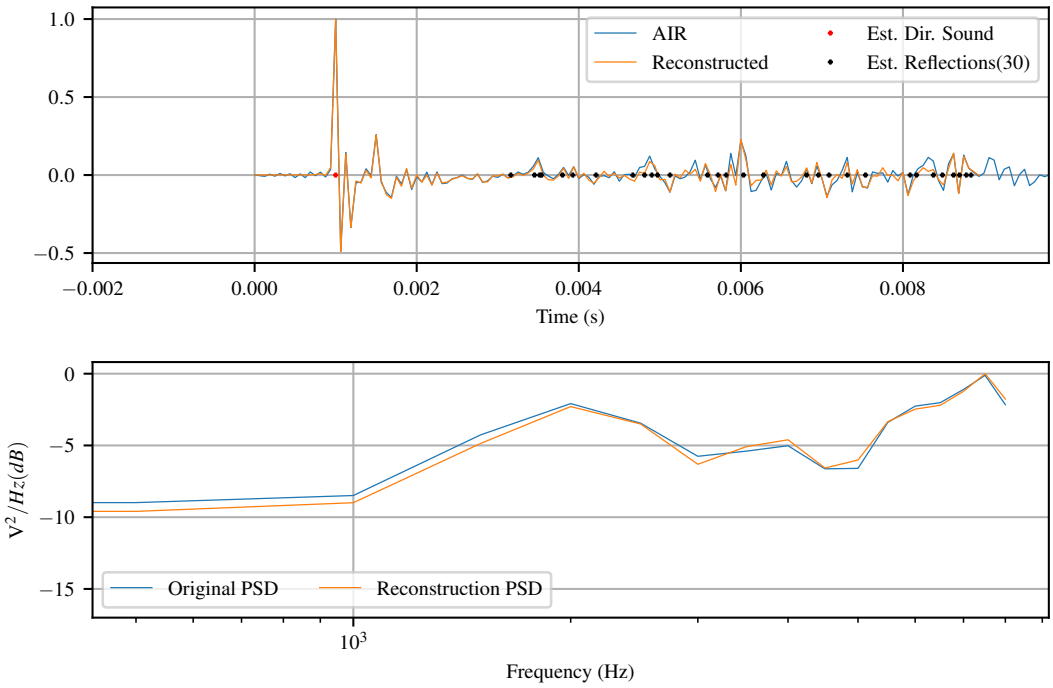
ering the effect of frequency-dependent absorption provides more sparse representations of the reflections. Not considering the effect causes multiple copies of the excitation to be used to model single reflections, which have been shaped by the effect of materials in the room. The uncertainty in the filters of the materials and the simplifications made with regards to scattering and angle of incidence however impact the reconstruction accuracy of the model, to the levels stated above.

The accuracy of modelling measured responses is reduced when compared to the case of simulated responses. For an illustrative comparison of the two cases, Figure 8.12 shows an example of each. The focus is on the first 8 ms after the arrival of the direct sound for visual clarity. The Figure shows that for the simulated response, the assumptions for the components of the process are well aligned with the observed samples, which leads to the perfect reconstruction of the AIR taps and of the PSD. The characteristics of the real AIR however are very different, when compared to the simulated response. Firstly, the excitation signal is much more complex and the energy in the samples is not as sparse. The diffuse energy between the early reflections, due to scattering, is present and the density of reflections from surfaces in the room is higher. Identifying reflections parameters in this scenario is therefore a more challenging task. Another significant challenge in the modelling of real AIRs is that the frequency-dependent absorptions in the room are not precisely known. Simulating AIRs models the frequency absorptions of sound by materials using the same information that is used to train the material detector DNN proposed in this Chapter. The surfaces of materials in the real world however have unique properties, which the proposed method estimates, based on *a priori* knowledge of the theoretical values of the absorption process.

Another significant difference between the results of modelling early reflections in simulated versus real responses is that for real responses the NPM and Itakura distances are not strongly and inversely correlated to the DRR. The Pearson correlation coefficient between the NPM and the DRR is $\rho = 0.052$ and the equivalent for the Itakura distance is $\rho = -0.008$, in contrast to the simulated AIR case, where both values were $\rho = -0.82$. The remarks in the previous paragraph have identified some fundamental difference in the

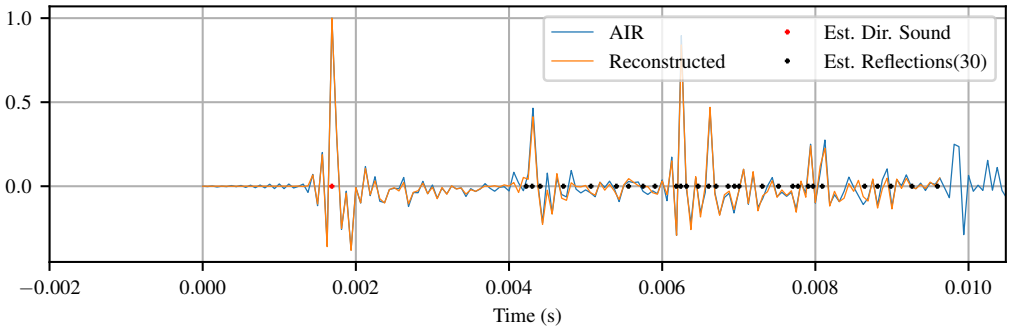


(a) Modelling of a simulated AIR.

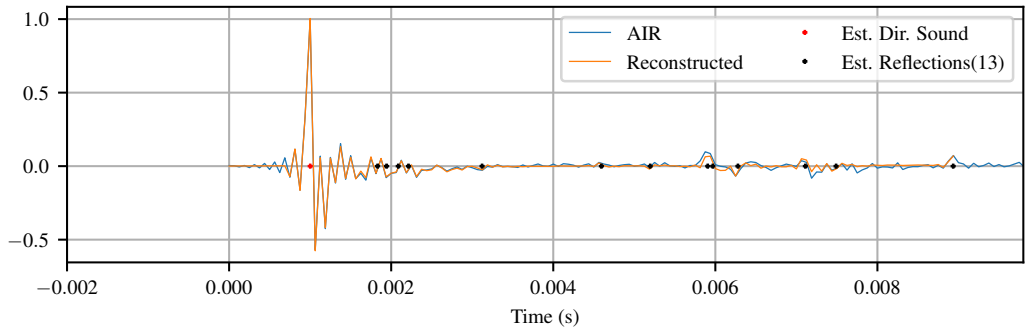


(b) Modelling of a measured AIR.

Figure 8.12: Comparison between the modelling of a simulated and a measured AIR in a meeting room, part of the ACE database [11]. Plots show the original AIRs and their reconstructions in the time domain and in terms of their PSD.



(a) Modelling an AIR with negative DRR (-4.80 dB) using the proposed method.

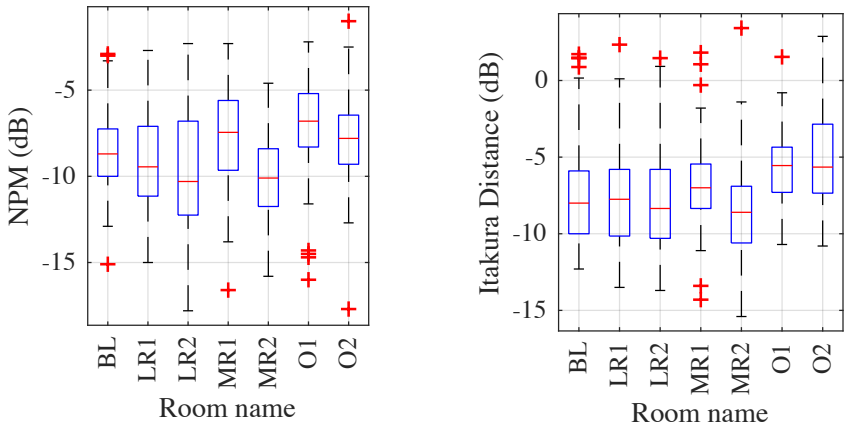


(b) Modelling an AIR with negative DRR (6.47dB) using the proposed method.

Figure 8.13: Modelling measured AIRs with negative and positive DRR using the proposed method. The two AIRs were measured in a lecture theatre, using the same microphone array, by moving the source and array between the two measurements.

modelling of real and simulated AIRs. To understand how the DRR affects this, Figure 8.13 shows 2 AIRs, measured in a lecture theatre, using the same microphone array, by moving the source and array between the two measurements. The two have DRR values of -4.80 dB and 6.47 dB. The observation from Figure 8.13 is that in the case of low DRR, the reflections are strong and their energy is significantly higher than the diffuse energy. The assumptions made by the proposed method therefore are reinforced by this, which enable the accurate modelling of the early reflections. The opposite is true for the high DRR case, where reflections are very low in energy, making their parameter estimation more challenging task.

As in the case of the experiments in Section 7.6, the evaluation of the proposed method is shown in terms of individual rooms. The results in Figure 8.14 show that the modelling accuracy depends on the room. Individual rooms show different median



(a) NPM (b) Itakura distance

Room Legend : BL:Building Lobby, L1: Lecture Room 1, L2: Lecture Room 2, M1: Meeting Room 1, M2: Meeting Room 2, O1: Office 1, O2: Office. Room Volume: BL: 72 m^3, L1: 200 m^3, L2: 360 m^3, M1: 92 m^3, M2: 250 m^3, O1: 47 m^3, O2: 48 m^3

Figure 8.14: Result of modelling early reflections in 700 AIRs, measured across 7 rooms. The result shows the NPM and the Itakura distance between the reconstruction of the FIR filter taps of the AIRs and the taps of the original AIRs. 100 AIRs were measured in each room.

values and distributions of the NPM and the Itakura distances between the AIRs and their reconstructions. The properties of individual rooms therefore is a factor that affects the accuracy of modelling measured AIRs using the proposed method. These properties involve the furniture and their materials, the position of the furniture and where and how the microphone array was positioned for the measurement. These factors are not part of simulating AIRs in *shoe-box* rooms, which leads to the differences in the results.

The next Section looks at how taps that cover the entire duration of the AIRs discussed above can be reconstructed. The method relies on the use of the parameters estimated by the proposed method for the early reflections and a set of established acoustic parameters for the description of the late reverberation.

8.6.3 Modelling of entire AIRs

The last experiment, which is described in this Section, studies the application of the estimated parameters in the reconstruction of the entire AIR. Stochastic models for describing the reverberant tail are a typical choice in the literature [13]. This experiment illustrates

the application of the early reflection reconstruction offered by the method proposed in this Chapter, as a way to complement these stochastic models for the reverberant tail in order to accurately reconstruct the FIR taps of entire AIRs.

For this experiment, the stochastic model used to represent the reverberant tail is based on Polack's model [10]

$$h_{\text{Polack}}(n) = \nu(n) \exp^{-\bar{\zeta} n T_s}, \tag{8.26}$$

where T_s the sampling period, $\nu(n)$ is sampled from $N(0,1)$ and $\bar{\zeta}$ is the average damping constant of (2.9) [10]. The resulting tail is filtered by an IIR filter, with numerator and denominator coefficients b and a respectively, to capture the spectral shape of the tail in the original AIR. In [23], the zeros and poles at receiver positions in reverberant rooms were analysed and it is shown that both represent properties of the environment. Poles describe properties of the enclosure, whereas zeros vary with position. An IIR filter with P zeros and R poles is therefore designed to convey these properties to the reverberant tail. The low orders of $P = R = 5$ are chosen to reflect the motivation for low-dimensional models. The order selection for IIR models for AIRs is discussed in [170]. The IIR coefficients are estimated using Prony's method [171] from the original AIR tail and are applied to Polack's model to give the filtered reverberant tail

$$h_{\text{tail}}(n) = \sum_{i=0}^{P} b_i h_{\text{Polack}}(n-i) - \sum_{j=1}^{R} a_j h_{\text{tail}}(n-j). \tag{8.27}$$

A cross-fading mechanism is used to avoid abrupt discontinuities at sample n_m, where the early reflections are mixed with the tail model. The mechanism is applied to the tail to allow it to fade-in to a maximum of unity at n_m and have symmetric values around it, giving the late reverberation model

$$h_{\text{late}}(n) = \begin{cases} 0, & \text{if } n < k_1 \\ \dfrac{h_{\text{tail}}(2n_m - n + k_1)}{h_{\text{tail}}(0)}, & \text{if } k_1 \leq n < n_m \\ \dfrac{h_{\text{tail}}(n - n_m - k_1)}{h_{\text{tail}}(0)}, & \text{otherwise.} \end{cases} \tag{8.28}$$

Early reflections are described by the parameters estimated by the method described in this Chapter, substituted in (8.4) as

$$h_r(n) = \sum_{i=2}^{\hat{D}} \hat{\beta}_i \hat{h}_e(n) * \frac{\sin\left[\pi(n-\hat{k}_i)\right]}{\pi(n-\hat{k}_i)} \; \mathfrak{F}^{-1}\left(\prod_{\theta=0}^{\Theta_{\text{tot}}-1} \Phi_\theta^{r_{\theta,i}}(K)\right) \tag{8.29}$$

and the direct sound by

$$h_d(n) = \hat{\beta}_1 \hat{h}_e(n) * \frac{\sin\left[\pi(n-\hat{k}_1)\right]}{\pi(n-\hat{k}_1)} \tag{8.30}$$

The early reflections and the late reverberation model are scaled according to the DRR values η_1 and η_2. They measure the energy ratio between the direct sound and the early reflections and the direct sound and the reverberant tail in the original AIR and impose the same ratios on its reconstruction. The complete model, reconstructing taps of the FIR filter representation of the AIR is given by

$$\hat{h}(n) = h_d(n) + \sqrt{\frac{\sum h_d^2(n)}{\eta_1 \sum h_r^2(n)}} h_r(n) + \sqrt{\frac{\sum h_d^2(n)}{\eta_2 \sum h_{\text{late}}^2(n)}} h_{\text{late}}(n). \tag{8.31}$$

The AIR measured in a meeting room is shown in Figure 8.15. The figure shows the original AIR and its reconstruction from the modelling discussed in the previous paragraph. Early reflections are considered up to $n_m T_s = 24$ ms, which is defined as the mixing time in [31]. The results show the accurate modelling of the early reflections and the addition of the cross-fading component models the diffuse energy between them, which is not captured by the sparse model. The late tail decay is described by the stochastic tail, using the T_{60} estimated using [30]. Preserving the DRR between the parts of the response aligns successfully the energy dynamics of the AIR between its three parts, the direct sound, the early reflections and the tail. The spectral modelling is also accurate in shape. The region below 1 kHz shows the highest modelling error with higher frequency regions being modelled more accurately. The NPM between the model and the original AIR is -8.67 dB and the Itakura distance is -9.70 dB.

Repeating the experiment of Section 8.6.2, now for the entire AIR, gives the NPM

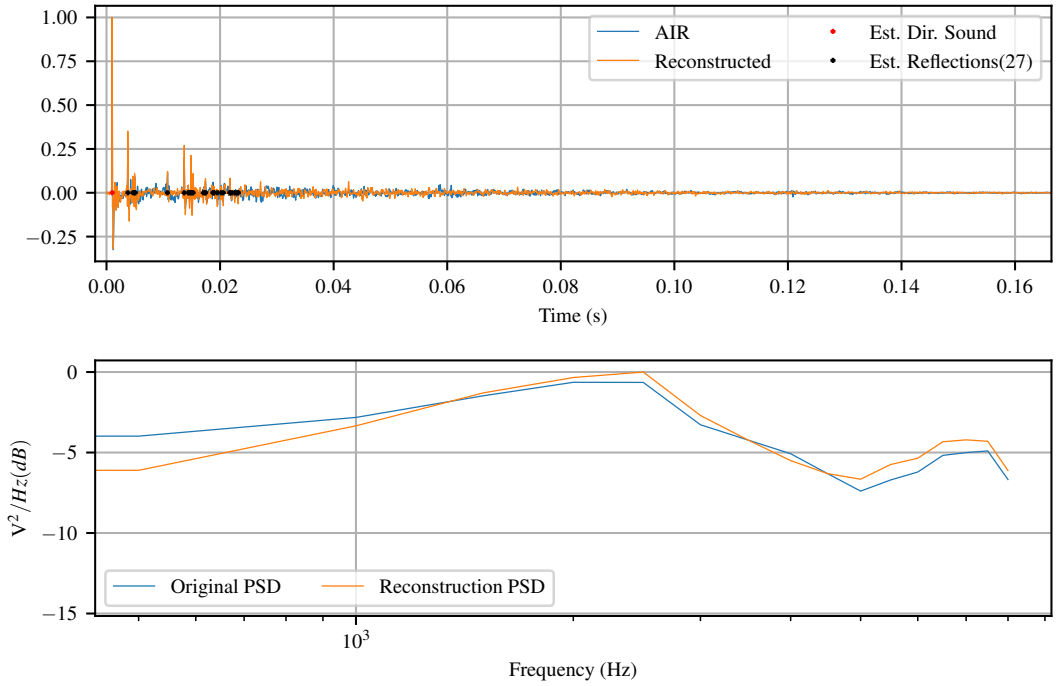
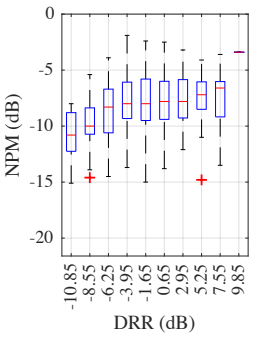
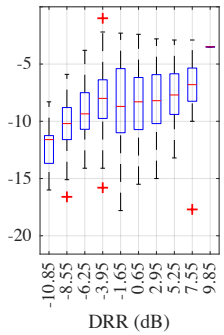


Figure 8.15: Modelling of an entire AIR measured for the ACE challenge in a meeting room. The model composes of a material-aware sparse parametric model for the early part and a stochastic model with IIR filtering for the late reverberation. The two parts are energy balanced using the DRR, to preserve the energy ratios between parts of the original AIR.

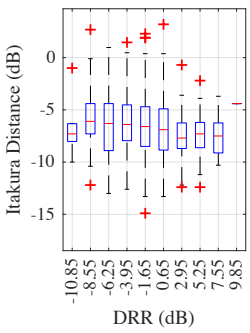


(a) Modelling of early reflections and late tail

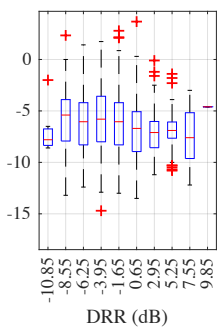


(b) Modelling of early reflections

Figure 8.16: NPM for modelling of measured AIRs using early and late part modelling.



(a) Modelling of early reflections and late tail



(b) Modelling of early reflections

Figure 8.17: Itakura distance for modelling of measured AIRs using early and late part modelling.

values of Figure 8.16 and the Itakura distance values of Figure 8.17. This experiment models 700 measured AIRs part of the ACE database [34]. The results are compared to modelling only the early part and show that the expansion of modelling the entire AIR, using the stochastic tail model, maintains the performance characteristics of the early part. The median difference in the NPM is 0.5 dB in favour of the modelling of the early reflections. The median difference in the Itakura distance is 0.01 dB, in favour of modelling the entire AIR. For the case of including the late reverberation model, the NPM values show less variability and even less correlation with the DRR. The number of outliers is also reduced.

This concludes the experiments for this Chapter and the next Section provides a discussion and a conclusion.

8.7 Discussion and conclusion

This Chapter proposed a novel method for modelling the early reflections in reverberant rooms. A novel method was also proposed for estimating the frequency-dependent absorptions of the materials of the surfaces in the room. Both methods start from an AIR as an FIR filter. The estimation of the frequency-dependent absorptions uses DNNs in order to detect the presence of materials in the acoustic environment, based on their sound absorption characteristics. Following this, early reflections are described by their ToAs, scales and their interaction with the present materials. This provides a set of parameters for the reflections that form a model, able to represent the early part of AIRs and reconstruct its FIR taps. As the parameter set of the representation is small and semantically meaningful, each parameter can be adjusted for controlled alterations of the acoustics of an environment, which finds applications in artificial reverberation [92]. As an extension of this, sampling the space of the parameters also generates new reverberation data, which finds use in data augmentation, as discussed in Part V.

The parametric model formed by the parameters estimated using the proposed method was evaluated in the experiments presented in this Chapter, using simulated and

measured AIRs. The performance of the algorithm was shown to be inversely correlated to the DRR of the simulated environment. For simulated responses, the NPM and Itakura distance between the reconstruction and the original response were as low as -40 dB. For measured responses, the task proved more challenging with the lower bounds of the two measures being -17 dB and -15 dB. The characteristics of individual real rooms were shown to be a significant factor in the accuracy of modelling early reflections. Further experiments have illustrated an application for the estimated parameters, which is to combine them with established methods for representing the tail and enable the reconstruction of entire AIRs. The relevant experiments showed that the NPM and Itakura distances between the entire AIR and its reconstruction remain at similar levels compared to modelling only the early part, with the median differences for the two measures between the two cases being less than 1 dB.

In conclusion, the proposed method for estimating parameters of the early reflections provides a low-dimensional and semantically meaningful representation of the early part of AIRs. The motivation for constructing this representation is the high dimensionality and the limited number of AIR measurements. These aspects limit the performance of state-of-the-art machine learning, as it was shown in Chapter 5. Crating this representation can enable machines to learn meaningful properties of the reverberation effect.

Part V

Generative Models for the Reverberation Effect

Chapter 9

Data Augmentation for Reverberant Environment Classification

9.1 Introduction

This Chapter presents a novel method for data augmentation for the training of classifiers of reverberant rooms. Data augmentation is the process of using existing training samples to artificially create additional ones. This increases the amount of available data for the training of classifiers [63]. The process preserves the classes of the original data, therefore the classifier is trained in this manner to be invariant to specific variations at the input and improve its generalisation.

Training data available for learning properties of the reverberation effect is often in the form of AIRs as FIR filters, measured in real rooms [17]. These AIRs are high dimensional, consisting typically of thousands of coefficients, and they are small in number. This limits the training of classifiers based on DNNs. The novel data augmentation method proposed in this Chapter starts from measured AIRs and generates additional artificial ones. This expands the training dataset without the need for further data collection, which is time-consuming and impractical. In the experiments shown in this Chapter, the proposed data

augmentation method improves the classification accuracy of reverberant rooms, when classified using the CNN-RNN designed for the task in Chapter 6.

The proposed method is based on learning how to artificially generate AIRs as if they were measured in a real room. A GAN is presented with real AIRs measured in one room, in order to learn how to do so. This enables the generation of many artificial AIRs from the same room, using only the limited number of measured ones. This is an alternative to the process of measuring many more AIRs in the room, by moving the source and receiver at various positions. A challenge to overcome during training is related to the motivation for this work, which is the high dimensionality of AIRs. This is overcome in this Chapter by using a proposed low-dimensional representation for acoustic environments. The representation describes sparse early reflection using the parameters estimated in Part IV and uses established acoustic parameters to represent the late reverberation. Creating a low-dimensional representation also allows for the evaluation of the generated responses and their distribution across a set of parameters relevant to the task. Evaluating the data generated by GANs is typically not straightforward, which is a drawback in their use [72]. In this work, the generated samples consist of a small and semantically meaningful parameter set, which allows for easier evaluation of the results. The generated AIRs find uses beyond the data augmentation of classifiers, such as artificial reverberation [92]. To illustrate the effectiveness of using the proposed low-dimensional representation of AIRs, experiments in this Chapter compare it with the use of the raw FIR taps.

The remainder of this Chapter is organised as follows: Section 9.2 discusses data augmentation for classification and Section 9.3 presents the proposed method for generating artificial AIRs for room classification training. The experiments in Section 9.4 present the results of the proposed method. Finally, Section 9.5 provides a discussion of the results of the experiments and a conclusion.

9.2 Data augmentation for classifier training

The supervised training of classifiers relies on the collection of labelled data, serving as the examples the classifier learns from. Recalling from earlier parts of this thesis, in Chapter 6, DNNs were presented with a set of AIRs in order to learn to discriminate between different rooms. In realistic scenarios, it is impossible for the training data to cover every point in the corresponding physical space. This means that unseen data will be presented to the classifier during inference when a speech recording is made at a source and receiver position not part of the original data collection. Covering every point of the room during data collection as a way to avoid this is not feasible. Substantially expanding the training data set for certain tasks such as room classification and SED, is challenging [172]. There exist however methods for increasing the amount of available training data without the need for additional data collection. This process is referred to as data augmentation [63].

The concept of data augmentation has been studied extensively in the literature in order to improve the accuracy of classifying audio signals. A very simple yet representative example of this concept is discussed in [173], which describes the task of detecting bird singing. In this example, any segment of audio containing bird singing would be a positive sample. Still, any mixture of two, or more, positive samples would also be positive. This simple mechanism of overlapping audio segments allows for the expansion of the amount of available training data without any additional data collection but with a simple overlap of two existing recordings. This *manual* method for data augmentation does not involve a statistical model but a simple logical reasoning and human understanding of the task. Other such methods discussed in the literature include time-stretching of segments [174], pitch shifting [175] and dynamic range compression [172]. A data augmentation method for audio data, which does not rely on such *manual* processes is proposed in this Chapter. The focus is the classification of reverberant environments and the method relies on generative models, able to generate additional artificial AIRs. These AIRs are used to improve the accuracy of a room classifier during inference.

The next Section discusses how DNNs are used to estimate generative models for different categories of reverberant rooms. Focus is given on how the networks are trained

to allow for efficient and effective model estimation. The issue of high-dimensionality of the FIR filter AIR representation is again a challenge to overcome and alternative solutions are given, based on the findings of this thesis.

9.3 Generative model estimation for reverberant rooms

The above Sections have discussed the motivation for estimating generative models that allow the generation of artificial AIRs corresponding to real reverberant rooms and how this can improve the accuracy and generalisation of room classifiers.

9.3.1 Estimation method

An introduction to generative models and their estimation using DNNs was presented in the background material of this thesis. Recalling from Section 3.1.4, a generative model represents the joint probability $P(\mathbf{x}, \mathbf{y})$, which is in contrast to classification DNNs that estimate the posterior $P(\mathbf{x}|\mathbf{y})$. Recent advancements in deep learning led to the proposal of alternatives to the traditional method of estimation of parametric model distributions. Two dominant methods exist, able to perform the estimation, GANs and VAEs. Both follow a similar formulation that uses back-propagation to train network layers, which are able to estimate the generative model by filtering noise drawn from a known prior. In the literature review conducted for this work, GANs have shown to be widely adopted in the field of audio processing across different tasks such as SED [53], speech recognition [176], speech enhancement [177] and dereverberation [178], [179]. Furthermore, variants of the original GAN in [68] exist, which can be adapted in the future to lead to more exciting applications of the method proposed in this Chapter, such as Conditional GANs [180], DualGANs [181] and many others. GANs are therefore chosen as the estimation mechanism for the generative models in this Chapter.

The rest of this Section discusses how the networks are trained to learn properties of the reverberation effect. Two options are explored, the first one uses the raw FIR taps of AIRs to train the GANs. The second, uses a proposed low-dimensional representation of

the AIRs based on the work of Part IV, to train the GANs.

9.3.2 GAN training

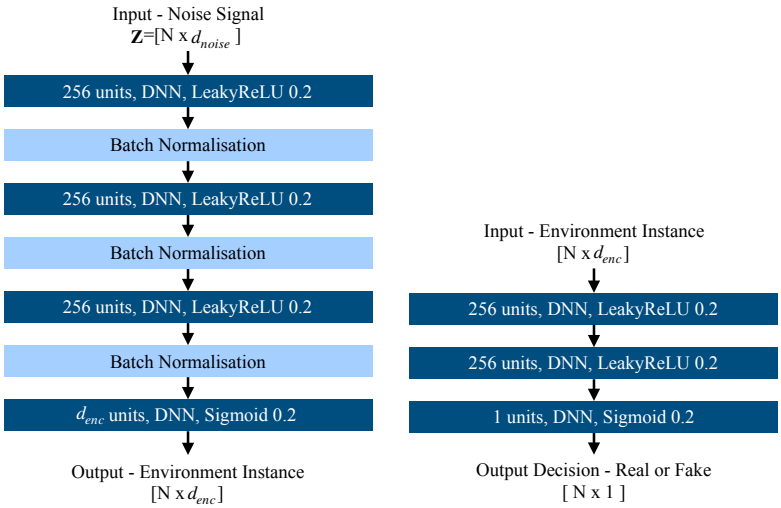
Recalling from Section 3.1.4, GANs are composed of two networks that are posed as adversaries. The two networks play the roles of the generator and discriminator [68]. The task of the discriminator $D(\mathbf{y};\theta_d)$ is to judge whether a given sample comes from the original data distribution or not. The task of the generator $G(\mathbf{y}|\mathbf{z};\theta_g)$, on the other hand, is to *fool* the discriminator into thinking that data samples it produces are originating from the original data distribution. $\mathbf{z}$ represents a random vector variable as $\mathbf{z} \sim N(\mathbf{0},\mathbf{I})$. The two networks hence play the minimax game given by (3.7), which minimises the Jensen–Shannon divergence between the generated data distribution and the true distribution, given an optimal discriminator. The training process for learning the parameters θ_d and θ_g, using the loss function of (3.7), relies on the iteratively performing the following steps:

1. The discriminator is trained for a number of steps k_d. Each step collects m_d samples from the training dataset and also draws m_d samples from distribution $P(\mathbf{z})$, which are then filtered by the generator network to give $G(\mathbf{y}|\mathbf{z};\theta_g)$. This provides a batch of m_d positive and m_d negative samples to the discriminator at each training step.
2. The generator is trained, trying to fool the discriminator that all samples it produces are real.

The training data passed to the discriminator in the first step are AIRs that were measured in a real room.

The networks are trained using back-propagation with Adam [59] as the optimizer. Inputs to the network are scaled to be within the range 0 and 1. The outputs are denormalised to restore the original scales. One normaliser-denormaliser pair is designed for each input-output neuron pair. The training is run for a total of 6000 epochs. For the reasons relating to stability discussed in [71], Additive White Gaussian Noise (AWGN) is added to the inputs of the discriminator with $\sigma = 0.1$.

The networks used in this work are shown in Figure 9.1 as the generator and discrim-



(a) Generator Network (b) Discriminator Network

Figure 9.1: Generator and discriminator networks of Generative Adversarial Networks (GANs) used for the generation of artificial AIRs.

inator of the GAN. A simple DNN architecture is used, composed only from FF layers. More complex architectures can be constructed that include convolutional and recurrent layers. The investigation of the benefits of using other types of layers, as well as techniques such as dropout, is reserved for future work. Given the small size of the network, LeakyReLU activations [63] are used instead of ReLUs as the activations, to counter issues that result from learning from gradients when the ReLU activations are 0. Batch normalisation is used in order to improve the training, as proposed in [182], by normalising the mean and standard deviation of activation.

Having defined the networks used as the generator and discriminator for the GANs used in this Chapter and their training method, the rest of this Section investigates how the training data is presented to the networks and the different choices for doing so.

9.3.3 GANs using the FIR filter taps of AIR

The GANs trained in this work are given a set of training AIRs measured in a specific room and learn how to create new AIRs as if they were measured in the same room. The networks use AIRs to learn properties of the room and also output artificially generated

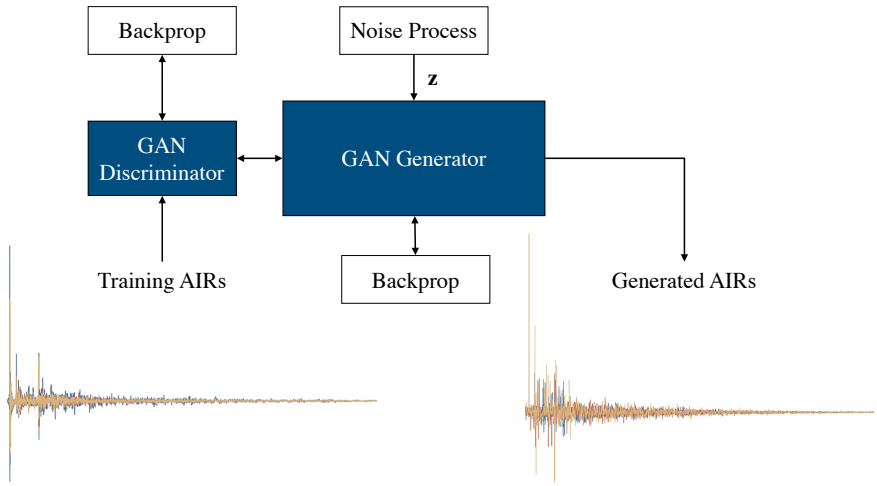


Figure 9.2: Training GANs to generate artificial AIRs, represented as FIR filters.

AIRs. This Sections investigates two choices for the way that AIRs are presented and outputted by the network.

The first choice is to represent AIRs using the taps of the FIR filters that describe them. This is the raw form in which AIR are typically measured [17] and distributed in [34]. This is shown in Figure 9.2. To represent the inputs and outputs of the networks, notation from Chapter 5 is reused. M AIRs, measured in real rooms, are represented by vectors $\mathbf{h}_m$, where $m \in \{1, 2, \ldots, M\}$. The ideal discriminator's behaviour, in this case, is therefore $D(\mathbf{h}_m) = 1 \; \forall \; m$. The ideal generator's behaviour is $D(G(\mathbf{z})) = 1$, where $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$.

9.3.4 GANs using a low-dimensional representation

An alternative to processing and generating AIRs as FIR filters is proposed in this Chapter. Describing the AIR as an FIR filer is a typical choice but leads to a sequence of thousands of taps to be processed by algorithms. An alternative low-dimensional representation of the acoustic environment therefore leads to fewer parameters to be processed, potentially improving the efficiency and effectiveness of training.

This Chapter proposes a representation that combines the early reflection parameters,

estimated using the method of Part IV, and a set of established parameters for describing late reverberation to represent the training AIRs. With the training AIRs represented in this space, the trained GANs will learn to model the distribution of each of the parameters, instead of how to model individually the thousands of FIR taps. Furthermore, generated AIRs will be in this low-dimensional space, reducing the complexity of the generator and discriminator. FIR filters that represent the generated AIRs can be constructed from this low-dimensional representation.

The rest of this Section describes how AIRs as FIR filters are used to estimate the parameters of the proposed low-dimensional representation. Also, the inverse process is described, which uses the low-dimensional representation to construct FIR filters. The two processes are respectively referred to as modelling and construction of the FIR filter taps of the AIR.

Low-dimensional representation

The aim of the data augmentation method proposed in this Chapter is to provide additional training samples to classifiers during their training, in order to improve their accuracy during inference. The aim is for the additional training samples to improve the classifier's generalisation, by presenting it with more examples of class invariant transformations of its inputs.

The task of room classification is to identify a known room at unknown source and receiver positions. As highlighted throughout this work, early reflections have a strong and distinct structure in AIRs, which is highly related to the source and receiver positions. Manipulating therefore the structure of early reflections corresponds to a manipulation of these positions. Generating additional AIRs that maintain the characteristics of the room but with different early reflection structures corresponds to taking more measurements from the same room and moving the source and receiver between the measurements. Randomly generating early reflections to do so does not guarantee the generation of reflection patterns that can be observed within the real room. Using a parametric representation of the early reflections, this Chapter uses GANs to learn the distribution of parameters

of reflections from data measured within the room that enables them to generate many artificial responses. These responses will be generated as if they were measured in the same room but with the source and receiver positions changing.

Chapter 7 has proposed a method for estimating the ToAs $\boldsymbol{\kappa}$ and scales $\boldsymbol{\beta}$ of D early reflections in an AIR and the excitation $h_e(n)$ that was used to measure it. This method allows for the early part of AIRs to be modelled using much fewer parameters than an FIR filter, by exploiting knowledge about the sparse nature of reflections. Modelling early reflections in this manner enables the reconstruction of the taps of the early part of the original AIR as

$$h_r(n) = \sum_{i=1}^{D} \beta_i \left[h_e(n) * \frac{\sin\left[\pi(n-k_i)\right]}{\pi(n-k_i)} \right]. \tag{9.1}$$

The experiments in Section 8.6.3 illustrated how this can be expanded to represent the entire AIR through the use of established parameters for describing reverberation. This is also done in this Chapter. Using Polack's model and the T_{60}, a decaying reverberant tail model is created, filtered by an IIR filter with numerator and denominator coefficients b and a respectively. This is expressed by (8.28) as the late reverberation model $h_{\text{late}}(n)$. This model for late reverberation and the reconstruction of the early reflections are scaled using two values for the DRR, η_1 and η_2. They respectively measure the energy ratio between the direct sound and the early reflections and the direct sound and the reverberant tail in the original AIR and impose the same ratios on its reconstruction. The direct sound is represented by $h_d(n)$. The FIR filter taps of the AIR are reconstructed from this representation using

$$\hat{h}(n) = h_d(n) + \sqrt{\frac{\sum h_d^2(n)}{\eta_1 \sum h_r^2(n)}} h_r(n) + \sqrt{\frac{\sum h_d^2(n)}{\eta_2 \sum h_{\text{late}}^2(n)}} h_{\text{late}}(n) \tag{9.2}$$

Based on the above, this Chapter proposes the use of a low-dimensional representation of AIRs to train GANs. The parameters forming the representation are estimated from the original AIR taps and are able to reconstruct them. The parameters are the following:

$\boldsymbol{\kappa}$: The ToAs of the D reflections up to 24 ms, estimated as proposed in Chapter 7.

$\boldsymbol{\beta}$: The scales of the D reflections with ToA up to 24 ms, estimated as proposed in

Chapter 7.

η_1, η_2: The two DRR values discussed above, measured from the original AIR using (2.10).

a, **b**: The coefficients of the IIR filter for the reverberant tail, estimated using Prony's method [171] from the tail of the original AIR.

T_{60}: The RT of the environment, estimated using [30].

All the above vectors are defined as column vectors. One column vector with fixed-length is used to represent each AIR. It is created using the above parameters and it is used for training the GANs. This column vector has a fixed-length and for AIR $\mathbf{h}_m$ it is expressed as

$$\tilde{\mathbf{h}}_m = \left[\, T_{60}, \eta_1, \eta_2, \mathbf{a}^T, \mathbf{b}^T, \mathbf{0}_{(D_{\max}-D)}, \boldsymbol{\kappa}^T, \mathbf{0}_{(D_{\max}-D)}, \boldsymbol{\beta}^T \, \right]^T . \tag{9.3}$$

The row vector $\mathbf{0}_{(D_{\max}-D)}$ is used to account for the fact that the number of early reflections detected in each AIR varies.

The excitation signal $h_e(n)$ in (9.1) accounts for the non-idealities in the method and equipment that was used to measure the AIR. Methods for its estimation are proposed in Section 7.3. Modelling the excitation is outside the scope of this Chapter and it is therefore not used for training the GANs. Including the excitation signal in the construction of the FIR filter introduces the non-idealities of the measurement method in the constructed taps. This is useful for creating AIRs that resemble real-life measurements. For the data generated in this Chapter for data augmentation, the excitation used for the construction of an artificial AIR is replicated from a randomly chosen training AIR. Using $h_e(n) = 1$ as the excitation represents the use of a source and receiver system with linear response and infinite bandwidth, which is unrealistic.

The IIR filter with coefficients **a** and **b** involves P zeros and R poles. It conveys information about resonances and the spectrum of reverberant speech recorded in the room [23]. Any poles that are part of the filter that are outside of the unit circle will lead to an unstable system. To prevent this, a zero-pole analysis is performed on generated values for **a** and **b** and any poles outside of the unit circle are removed. With the focus being on low-dimensional representations, the small values of $P = R = 5$ are chosen. The

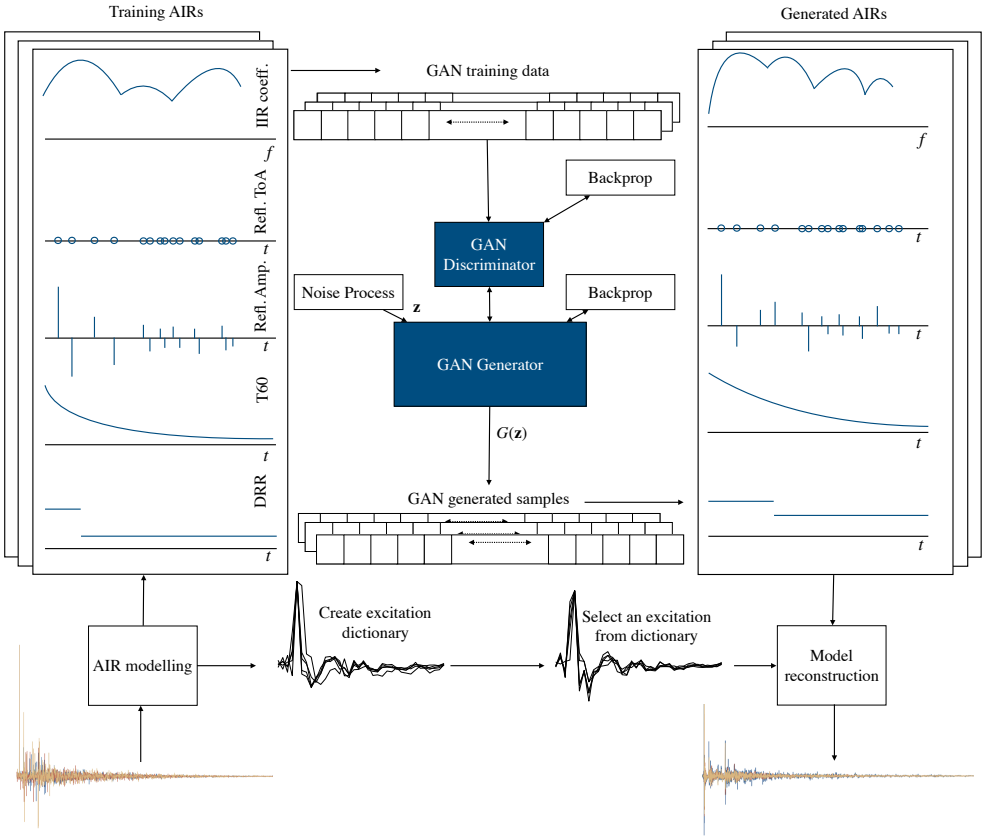


Figure 9.3: Training GANs to generate AIRs using the proposed low-dimensional representation of the training AIRs. The representation consists of parameters describing the early reflections and a description for the effect of late reverberation. The parameters of the generated AIRs then construct the taps of FIR filters, which are used to generate artificial training data for room classification training.

order selection for IIR models for AIRs is discussed in [170].

The overall process that is described in this Section is summarised in Figure 9.3. The following Sections describe experiments that evaluate the effectiveness of each of the two methods for generating AIRs for data augmentation for the training of room classifiers. The experiments will first present and analyse the generated AIRs. Their usefulness will be measured in terms of the gains in accuracy provided for the task of room classification.

9.4 Experiments

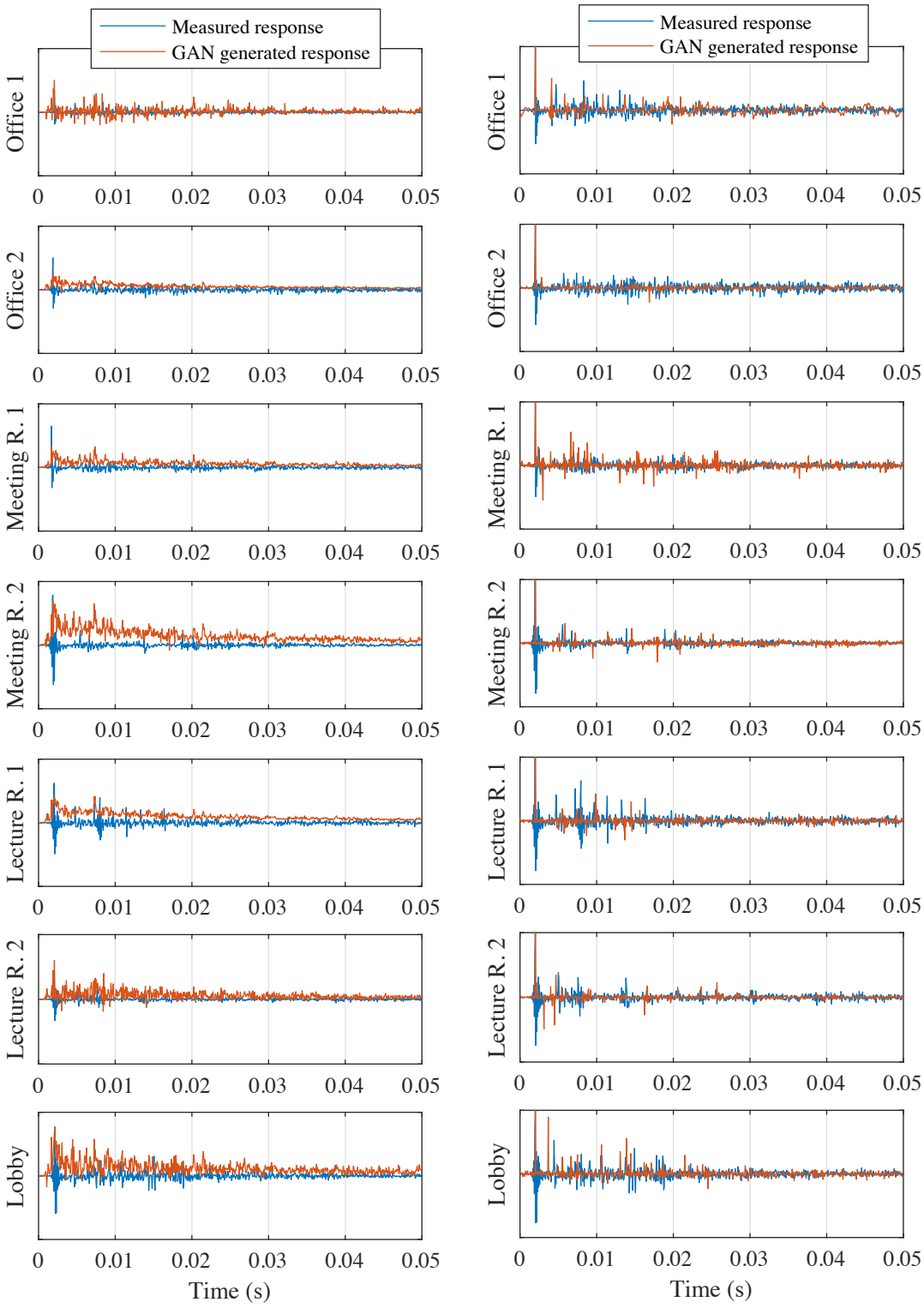
This Chapter proposes a novel method for data augmentation for the training of DNNs for room classification. The method relies on the training of GANs, which are used to generate artificial AIRs that are used for the training of the classifiers. The experiments described in this Section illustrate the data generated by the GANs and room classification experiments evaluate their efficacy in data augmentation. To highlight the usefulness of the proposed method, it is compared to the use of the raw taps of FIR filters for the training of GANs.

The dataset used to train the GANs is a set of AIRs provided with the ACE challenge database [34]. A total of 658 responses are used to train the GANs, split evenly across 7 rooms. The AIRs are padded to a duration 2.1 s, the duration of the longest AIR in the training dataset. All data is downsampled to 16 kHz.

9.4.1 AIR generation

The first experiment investigates the use of the raw FIR taps of AIRs for the training of GANs. Information about the number of parameters needed to train the generator and discriminator networks for this task are given in Tables 9.1a and 9.2a. One GAN for each of the 7 rooms is trained. One artificial and one real AIR for each room are shown in Figure 9.4a.

The process described above is repeated, with the GANs now trained using the proposed low-dimensional representation of AIRs. The number of parameters needed to train the generator and discriminator networks for this task are given in Tables 9.1b and 9.2b. The networks are the same in this case as for the case of using the raw FIR taps. The input-output dimensionality though has changed. The total number of parameters of the networks is now reduced by 98% and 99% for the generator and discriminator networks respectively. The generated data using the resulting networks contain information about the reflections at the receiver and the late reverberation effect. They are used to construct a set of AIRs as FIR filters. In Figure 9.4b, one constructed artificial response per room



(a) GANs trained using FIR taps of AIRs

(b) GANs trained using the proposed low-dimensional representation. The generated AIRs in this representation are used to construct the taps of FIR filters, shown in the plots.

Figure 9.4: Comparison between artificially generated and measured AIRs. Artificial AIRs are generated by GANs, trained using AIRs measured in the real rooms.

Layer	Param. count	Input Dim.
FF	5,376	20
Batch N.	1,024	256
FF	65,792	256
Batch N.	1,024	256
FF	65,792	256
Batch N.	1,024	256
FF	8,544,736	256
Total	8,684,768	33,248

(a) Generator network.

Layer	Param. count	Input Dim.
FF	8,511,744	33,248
FF	65,792	256
FF	257	256
Total	8,577,793	1

(b) Discriminator Network.

Table 9.1: Parameters per layer for a GAN, trained using FIR taps of AIRs.

Layer	Param. count	Input Dim.
FF	5,376	20
Batch N.	1,024	256
FF	65,792	256
Batch N.	1,024	256
FF	65,792	256
Batch N.	1,024	256
FF	43,690	256
Total	183,722	156

(a) Generator network.

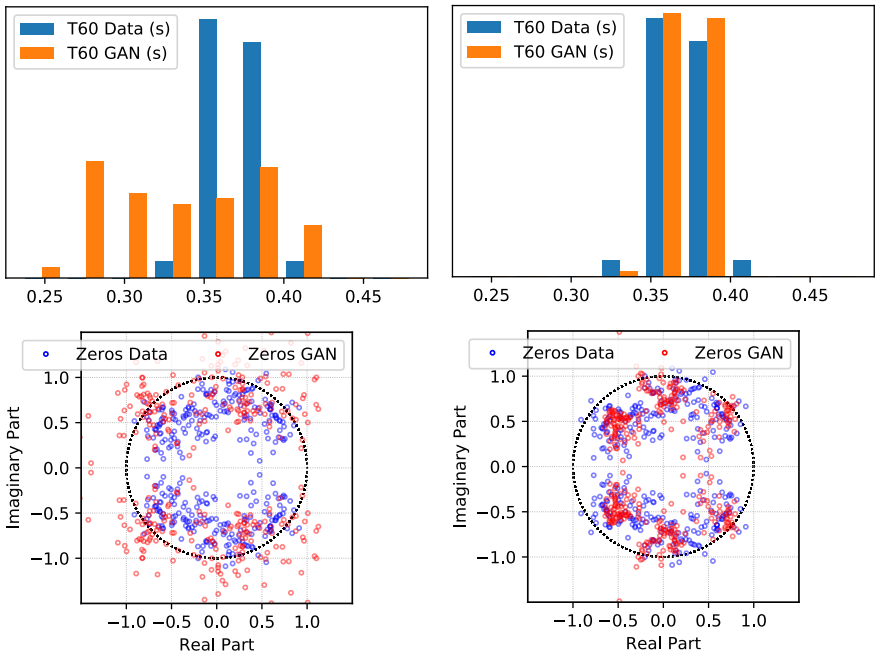
Layer	Param. count	Input Dim.
FF	43,776	156
FF	65,792	256
FF	257	256
Total	109,825	1

(b) Discriminator network.

Table 9.2: Parameters per layer for a GAN, trained using AIRs in the proposed low-dimensional representation.

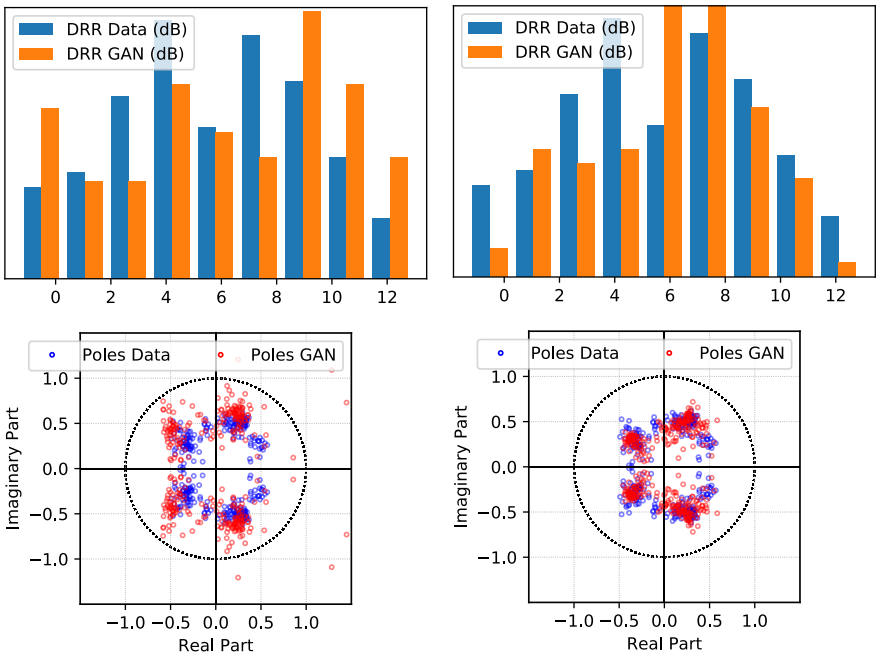
is visualised along with a measured one. The measured responses are shown as the blue lines and are used as a reference. They are the same as in Figure 9.4a.

Using the proposed low-dimensional representation for training allows for the analysis of the generated AIRs with regard to the parameters that form the representation. One of the critically discussed issues in the literature with regard to the training of GANs is the lack of established methods for evaluating the generated data [72]. The results can always be evaluated for their usefulness for data augmentation but a way to evaluate the generated samples directly saves unnecessary training times. Indeed, in this work, it would also be of interest to evaluate how realistic the generated responses are. Evaluating realism is difficult and it is very hard to quantify or even precisely define. Using the proposed representation as the space for GANs however allows for an inspection of semantically meaningful properties of the generated environments by humans. For instance, it allows for the visualisations of Figure 9.6 to be made. They clearly show how the distributions of



(a) Epoch 80 (b) Epoch 960

Figure 9.5: Evolution of the distribution of generated T_{60} values and the frequency of the zeros of the generated IIR filters during the training of a GAN, using the proposed low-dimensional representation of AIRs.



(a) Epoch 80 (b) Epoch 960

Figure 9.6: Evolution of the distribution of generated DRR values and the frequency of the poles of the generated IIR filters during the training of a GAN, using the proposed low-dimensional representation of AIRs.

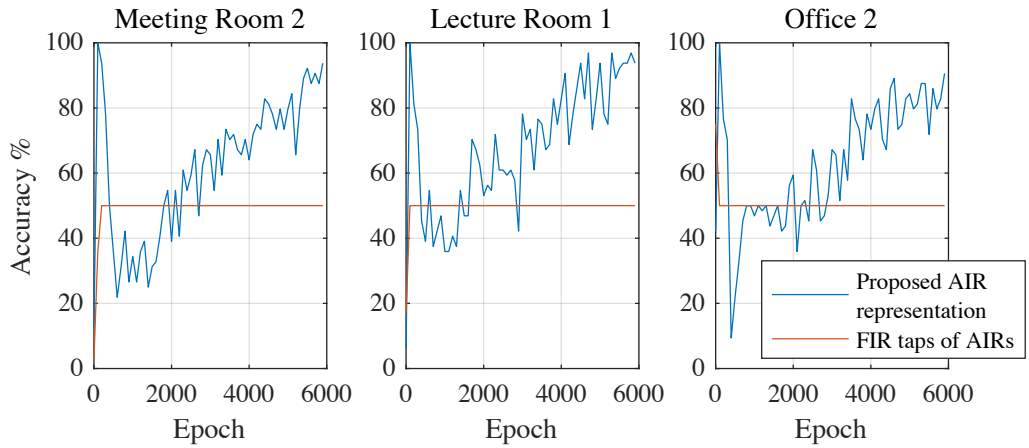


Figure 9.7: Comparison of the accuracies of GAN discriminators for the case of training using AIRs in the proposed low-dimensional representation and for the case of using the taps of the FIR filters of AIRs.

the T_{60}, DRR and the zeros and poles of the generated AIRs are distributed as the training of a GAN progresses. Observing the plots shows that the distribution of the parameters starts from a random state, which bears no resemblance to the one of the training data. As the training of the GAN progresses, the distribution of the generated data becomes very similar to the original. This is positive evidence that the distribution of these parameters is realistic as it follows the one in the training data, which is measured in the real rooms.

Using the raw FIR taps of AIRs directly to train a GAN provides no semantically meaningful information about the acoustic environment, in contrast to using the proposed representation. The results of Figure 9.4a is the only evidence for the quality of the results and it is inspected directly to analyse them. What can be observed from looking at the real AIRs, given by the blue lines in the Figure, is that they are composed of a direct path, sparse reflections in the early part and a decaying envelope. The same does not apply for the generated AIRs however that resort to tracking the overall energy envelope as an approximation to the overall shape. The discriminator of Figure 9.1b is therefore *fooled* by a simple imitation of the energy envelope of the inputs into believing that they are real AIRs. In reality, the generated responses fail to capture the sparsity of the early part. The opposite is true for the case of training GANs using the proposed representation, as shown in Figure 9.4b. The sparse nature of the early part is well captured and so is the

tail.

Probing into the process of training the network reveals important information, which explains the above observations. Figure 9.7 shows the accuracy of the discriminator for the cases of training the GANs for 3 rooms, using each representation. The case of training GANs directly using FIR taps shows the discriminator being unable to discriminate between real and fake samples after a small number of epochs. The accuracy plateaus to 50%, which indicates that the discriminator is making a random decision between *real* and *fake*. A weak discriminator cannot yield a stronger generator as Nash equilibrium is reached at this point [68]. The opposite is true for the case of using the proposed representation, as the discriminator's accuracy almost continuously increases and reaches values as high as 90%. This is attributed to the high dimensionality of the raw AIRs. This high dimensionality causes the first dense layer of the discriminator network of Table 9.2a, relating to discriminating between real and fake raw AIR FIR taps, to act as a bottleneck layer, responsible for heavily compressing information. Moving from an input with dimensionality 33248 down to 256 neurons directly through one hidden FF layer is not advised [47]. However, in this case, the GAN has scaled-up to more than 17 million parameters and the small amount of training data does not allow for the training of very large networks that would result from adding more layers. This actually brings this work back to its original motivation, which was high-dimensionality and the lack of large data availability. This reinforces the need for low-dimensional and informative representations for AIRs, such as the one proposed in this Chapter.

The experiments above have discussed how GANs are trained to generate AIRs for a set of rooms. One GANs is trained for each of the 7 rooms, part of the training data. Each network then generates a set of artificial responses as if they were measured in the real room. The following experiments will show how the generated responses are used as a data augmentation dataset to tackle the small availability of training data in order to improve the accuracy of a DNN room classifier.

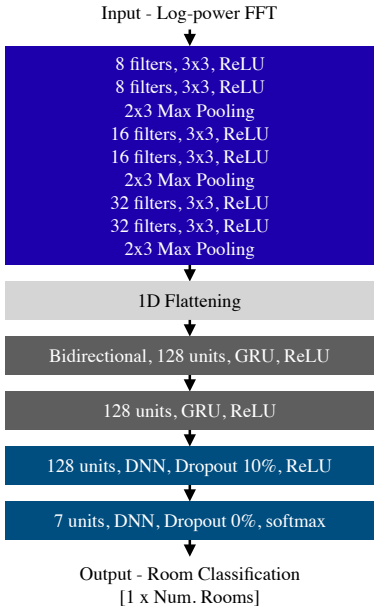


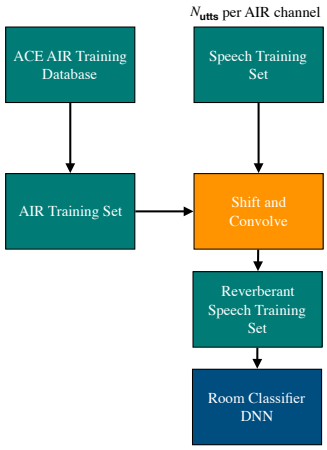
Figure 9.8: CNN-RNN models for room classification from reverberant speech input.

9.4.2 Data augmentation for DNN room classifiers

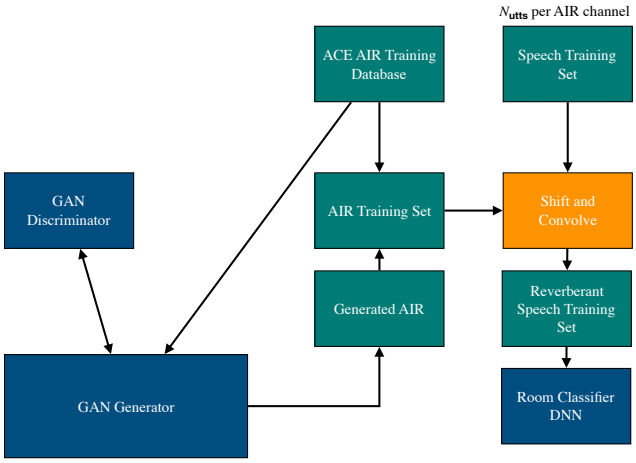
Room classification using state-of-the-art classifiers was investigated in Chapter 6. DNNs were used to classify a reverberant speech signal in terms of the room it was recorded in. The investigation has shown that the limited availability of AIRs and their high-dimensionality limit the performance of classifiers. This Chapter proposes novel methods for increasing the availability of training data in the form of AIRs, with the aim to increase the accuracy of DNN classifiers. The experiments above have shown how GANs are used to create artificial AIRs, given a set of real ones. These AIRs are used as part of the proposed data augmentation method.

Classifier DNN

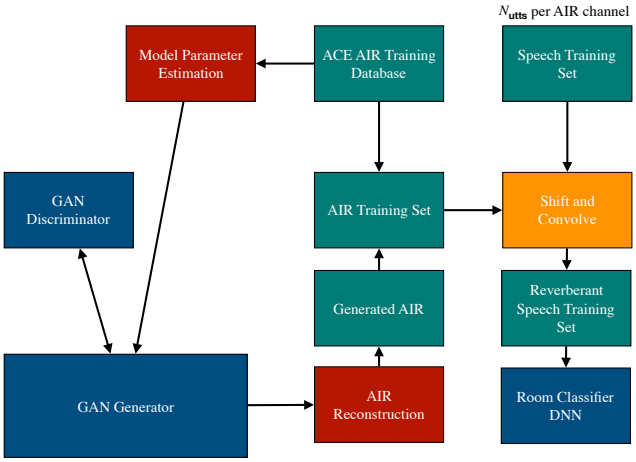
Chapter 6 has compared 4 candidate DNN architectures as classifiers for reverberant rooms. The DNN as a classifier in this experiment is taken as the best performing model from that comparison. The model is a CNN-RNN and it is shown in Figure 9.8. The training process remained unchanged from the one in Chapter 6.



(a) Classifier training without data augmentation.



(b) Classifier training with data augmentation using GANs trained by AIRs as FIRs filters.



(c) Classifier training with data augmentation using GANs trained by AIRs in the proposed low-dimensional representation.

Figure 9.9: Evaluation configurations for the effect of two different data augmentation strategies for the training of room classification DNNs.

The network is trained and evaluated in 3 configurations, shown in Figure 9.9. In Figure 9.9a the baseline is shown, which involves no data augmentation. It only uses measured AIRs from the ACE database [122] for training. Figure 9.9b shows the configuration where data augmentation is done using artificial AIRs generated by GANs trained using the raw FIR taps of AIRs. Finally, Figure 9.9c shows data augmentation done using the method proposed in this Chapter. Each network is evaluated based on its accuracy on a test set, which is discussed below. The networks are all trained individually on 16 different machines with NVIDIA Tesla K80 GPUs to average the effect of different initialisations.

Training and test data

The ACE database AIRs [122] are used for this experiment. They are segmented to reserve a test set prior to training. The ACE database consists of a total of 700 AIR, recorded in 7 rooms, as shown in Table 5.2. The 42 AIRs, which were recorded using the *Mobile* microphone array are reserved for testing. The remaining 658 AIRs are all used to train the relevant GANs and artificially create further AIRs for data augmentation. They contain an even number of 94 AIRs per room. The proposed data augmentation method involves the training of 1 GAN for each of the 7 rooms. Each GAN is trained using 94 AIRs and it is used to generate an additional 100 AIRs. This results in an additional 700 AIRs in total, doubling the size of the ACE database. The training data along with the artificially created AIRs, form the training set in the proposed method.

The experiment is investigating the classification of reverberant speech in terms of the room where the recording took place. All training AIRs are therefore convolved with 20 speech utterances each of length 5 s, taken from the TIMIT database. This is identical to the process of Chapter 6. Train and test speech and speaker databases are separated and are not mixed. Each utterance contains only one speaker. Convolving new speech samples with the data augmentation AIRs will introduce an additional variable in the comparison of the results. Therefore to avoid this, the same exact speech utterances convolved with the measured AIRs for each room are convolved with the data augmentation AIRs for the corresponding rooms. The 42 test AIRs are convolved with 10 utterances each, again

of length 5 s. The test and train reverberant speech is consistent throughout all the experiments and the only variable is the addition of the data augmentation AIRs. All data is sampled at 16 kHz.

The segmentation of the available data into test and training sets is discussed above. Neither the speaker position, the position of the microphone array or the array itself, all used to construct the test data, were presented to the classifier during training. The artificial AIRs, generated by the GANs, were generated as if they were measured in the same rooms as the training data but at different positions. The addition of these AIRs generated by the GANs aims to improve the classification test accuracy. Data augmentation performed in this fashion provides class invariant transformations of the training data to the classifier. This experiment therefore evaluates the improvement in the generalisation of the classifier offered by the proposed data augmentation method.

Results

The results of evaluating the proposed method are shown in Figure 9.10 and Table 9.10, in terms of the classification accuracy on the test set. The results show that the proposed method outperforms the baseline in all runs. The median accuracy of the baseline is 89.4%, the proposed method's is 95.5% and the AIR tap based method's is 87.15%. The proposed method therefore increases the accuracy of the room classifier. The increased accuracy is not attributed to an increase of the speech data as the exact same speech samples were used in all 3 cases. The use of the high-dimensional raw AIR taps proved even less effective than the baseline and the trained GANs involved a total of around 17 million parameters. The proposed domain increased the classification accuracy while using only 0.29 million parameters. To understand how the proposed data augmentation method affects the training, the training and validation losses of the trained DNNs are shown in Figure 9.11. The baseline training loss, which uses no data augmentation, shows that after 10 epochs the model starts to overfit. The training loss starts to substantially decrease and the validation loss increases. When using the proposed data augmentation, the validation loss continues to follow a decreasing trend for longer and the training loss

Run	None	Proposed	AIR FIR taps
1	89.3	**96.9**	88.6
2	88.6	**97.1**	97.4
3	92.1	**95.0**	86.9
4	90.2	**95.2**	89.3
5	89.0	**97.1**	85.0
6	89.5	**95.5**	85.0
7	82.4	**92.4**	86.9
8	93.6	**96.7**	89.5
9	88.6	**96.7**	87.4
10	89.3	**92.1**	83.3
11	91.1	**94.3**	89.5
12	93.6	**94.0**	85.2
13	91.0	**94.8**	84.5
14	87.9	**95.5**	86.4
15	82.6	**95.7**	87.4
16	90.5	**95.7**	88.6
Median	89.4	**95.5**	87.2

Table 9.3: Accuracy of room classification DNNs for the cases of using different representations of the AIR for data augmentation. Different runs indicate different initialisations and hardware.

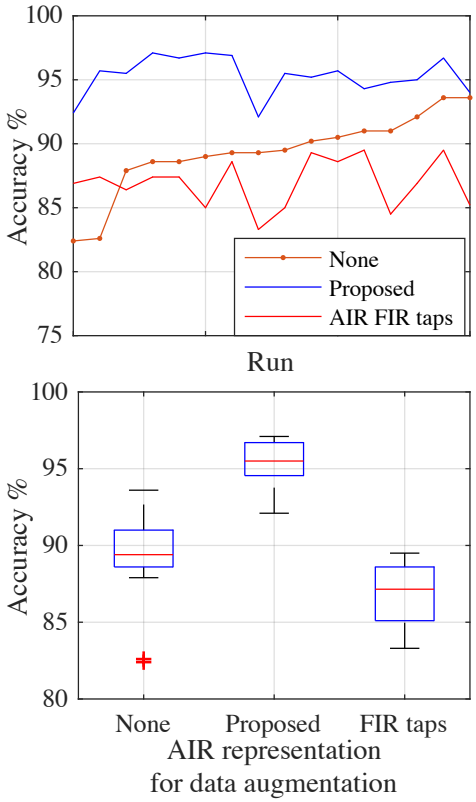


Figure 9.10: Accuracy of room classification DNNs for the cases of using no data augmentation (baseline), the proposed data augmentation method and data augmentation using the raw FIR taps of AIRs. Top results are sorted from worst to best with regards to the baseline. Individual runs indicate training on a different machine, all with NVIDIA Tesla K80 GPUs.

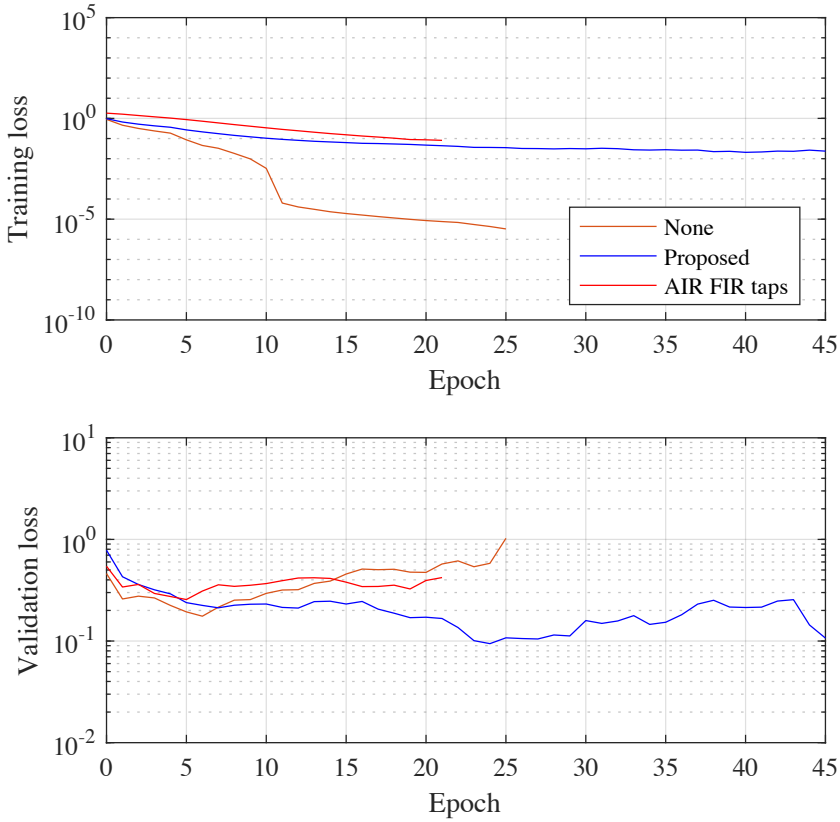


Figure 9.11: Comparison of the training and validation losses for a room classification DNN for the cases of using no data augmentation, the proposed data augmentation and the data augmentation using FIR taps of AIRs. Losses are smoothed using a moving average window of 10 epochs.

is approximately monotonically decreasing.

This analysis of the results of the experiments concludes this Section. The final Section will review the contributions of this Chapter, offer a discussion of the results of the experiments and provide a conclusion.

9.5 Discussion and conclusion

This Chapter has proposed a novel method for data augmentation for the training of DNNs for the task of room classification. The proposed method relies on the training of GANs, using AIRs in a proposed low-dimensional representation. The representation combines parameters of the early reflections, estimated using methods proposed in this

thesis, and established parameters for late reverberation. The GANs are used to create artificial AIRs from a set of known rooms. The proposed method was compared to the alternative of training GANs using the raw AIR taps from their FIR filter representation. The proposed method enabled GANs to generate artificial responses with realistic features, able to capture the sparse properties of the early reflections and the decaying tail. The opposite was true for the alternative.

The generated responses are used for data augmentation in order to improve the accuracy of room classifiers. In the experiments presented in this Chapter, using the proposed method increased the accuracy of classifying reverberant speech, when compared to the case of using no data augmentation. Inspecting the training and validation losses of the trained classifiers showed that without data augmentation the classifiers overfitted the training data. Adding the data augmentation samples improved generalisation, increasing the classification accuracy during inference. This data augmentation method improves generalisation by providing a meaningful and realistic interpolation of the available AIRs in a low-dimensional manifold of the reverberation effect. What makes this technique highly effective is that the learning is done using a representation that consists of a small number of parameters, which are important to the reverberation effect.

The training of GANs proposed in this Chapter uses AIRs measured in a room in order to create a number of artificial AIRs as if they were measured in the same room. This process finds applications beyond room classification. Artificial reverberation applications [92] can benefit from such approaches, where a number of artificial environments with specific properties can be created by training GANs using a specific modality of acoustic environments. For instance, providing GANs with enough AIRs from many concert halls will enable it to learn to generate many more artificial AIRs from many artificial concert halls. The possibilities for such methods are numerous.

In the experiments presented in this Chapter, the proposed method for data augmentation was effective in generating realistic AIRs and increasing the accuracy of DNN room classifiers. The results show that the accuracy increased from 89.4% to 95.5%, when compared to the case of using no data augmentation.

Part VI

End Matter

Chapter 10

Conclusion

10.1 Summary

The reverberation effect is an important part of our everyday listening experiences. Properties of the effect allow us to understand the properties of the rooms, simply by listening to reverberant sounds. The effect therefore contains cues for these properties that humans exploit to make certain distinctions. This thesis investigated how machines can do the same and use information from the reverberation effect to better understand the world around them. Issues relating to the representation of the acoustic environment have been addressed as an important difficulty to overcome in the process of teaching machines how to understand it.

The work done in this thesis can be summarised by the following contributions

- Proposed feature domains for the classification of acoustic environments. The domains were constructed using subsets of acoustic parameters. Also, novel results were provided on the comparison of the discriminative properties of the parameters on a set of classification tasks, by considering a number of classifiers. (Chapters 5)
- Proposed methods for the training of generalisable DNNs for the task of end-to-end room classification. A CNN-RNN architecture was proposed for the task. The investigation provided novel results about the performance of DNNs on the task of discriminating between different rooms and as to the features learned by the

classifiers. (Chapter 6)

- Proposed a novel method for estimating the parameters that describe the early reflections in an AIR and the excitation that was used to measure it. (Chapter 7)
- Proposed a novel method for estimating the frequency-dependent absorption by the surfaces of materials in a room by analysing an AIR, measured in the room. Proposed a novel method for improving the modelling of early reflections, using the knowledge about absorptions by the surfaces of materials in the room. (Chapter 8)
- Proposed a novel method for data augmentation for the training of DNN room classifiers based on GANs. This method improved the generalisation of the trained classifiers. The trained GANs were presented with a set of AIRs that were measured in a room and learned how to produce artificial ones as if they were measured in the same room. (Chapter 9)

10.2 Future work

The results of the analysis of the work and the contributions described above pave new avenues of research, which include the following:

AIR estimation. Investigate how DNNs can be used to estimate the parameters of the low-dimensional representation proposed in this thesis to described the AIR, directly from reverberant speech. This will be an alternative to the FIR filter estimation for the AIR, which is time-consuming and suffers from the high-dimensionality of the FIR representation [10].

Transfer learning. Training DNNs to classify reverberant speech inputs has shown that the derived feature-maps from convolutional layers are highly relevant to the decay of sound energy at different frequencies. This decay is measured by the FDRTs, which are parameters that can be measured from AIRs. The task of classification and parameter estimation [129] can therefore interact and transfer learning can be used between the two to improve their performance. For instance, the ground truth values of FDRTs can be used

to pre-train layers of room classification DNNs to improve the accuracy and convergence during training.

GANs for reverberant environments. The use of GANs to generate artificial AIRs in the future can include information about the materials in the room and encode their interaction with acoustic reflections, as discussed in Chapter 8. Furthermore, many variants of GANs exist in the literature, which offer exciting opportunities for research on the effect of reverberation. One exciting avenue of research would be to use cross-modal data generation [183] and investigate how reverberation can be generated from an image of a room for example. Another possibility would be to even consider a text-to-reverberation GAN, such as the text-to-photo GAN in [184].

Many exciting avenues of research arise from the work presented in this thesis, which promise advancements in the understanding of reverberation both by machines and humans.

Bibliography

[1] C. L. Christensen, G. Koutsouris, and J. H. Rindel, "Estimating absorption of materials to match room model against existing room using a genetic algorithm," in *Proc. of Forum Acusticum*, Krakow, Poland, 2014, pp. 7–12.

[2] C. Papayiannis, C. Evers, and P. A. Naylor, "Discriminative feature domains for reverberant acoustic environments," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, Louisiana, USA, Mar. 2017, pp. 756–760.

[3] ——, "Sparse Parametric Modeling of the Early Part of Acoustic Impulse Responses," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 708–712.

[4] ——, "End-to-end discriminative models for the reverberation effect," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (to be submitted)*, 2019.

[5] ——, "Data augmentation using GANs for the classification of reverberant room," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (to be submitted)*, 2019.

[6] C. Papayiannis, J. Amoh, V. Rozgic, S. Sundaram, and C. Wang, "Detecting Media Sound Presence in Acoustic Scenes," in *Proc. Conf. of Intl. Speech Commun. Assoc. (INTERSPEECH)*, Hyderabad, India, Sep. 2018, pp. 1363–1367.

[7] H. Kuttruff, *Room Acoustics, Fifth Edition*, English. CRC Press, 2009.

[8] M. Ermann, *Architectural Acoustics Illustrated*, English. John Wiley & Sons, 2015.

[9] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 114–126, Nov. 2012.

[10] P. A. Naylor and N. D. Gaubitch, Eds., *Speech Dereverberation*. Springer, 2010.

[11] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "The ACE Challenge - corpus description and performance evaluation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2015.

[12] A. Billon and J.-J. Embrechts, "Numerical evidence of mixing in rooms using the free path temporal distribution," *J. Acoust. Soc. Am.*, vol. 130, no. 3, pp. 1381–1389, 2011.

[13] A. Lindau, L. Kosanke, and S. Weinzierl, "Perceptual evaluation of model and signal-based predictors of the mixing time in binaural room impulse responses," *J. Audio Eng. Soc. (AES)*, vol. 60, no. 11, pp. 887–898, Dec. 2012.

[14] T. Paatero and M. Karjalainen, "New digital filter techniques for room response modeling," in *Proc. Audio Eng. Soc. (AES) Conf.: Architectural Acoustics and Sound Reinforcement*, St. Petersburg, Russia: Audio Engineering Society, Jun. 2002.

[15] G. Vairetti, T. van Waterschoot, M. Moonen, M. Catrysse, and S. H. Jensen, "Sparse Linear Parametric Modeling of Room Acoustics with Orthonormal Basis Functions," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Sep. 2014, pp. 1–5.

[16] J.-M. Jot, "An analysis/synthesis approach to real-time artificial reverberation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, Mar. 1992, pp. 221–224.

[17] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proc. Audio Eng. Soc. (AES) Convention*, Feb. 2000, pp. 1–23.

[18] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Process.*, vol. 43, no. 12, pp. 2982–2993, Dec. 1995.

[19] S. Subramaniam, A. Petropulu, and C. Wendt, "Cepstrum-based deconvolution for speech dereverberation," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 392–396, 1996.

[20] J. Mourjopoulos, "On the variation and invertibility of room impulse response functions," *J. of Sound and Vibration*, vol. 102, no. 2, pp. 217–228, 1985.

[21] M. Karjalainen, T Paatero, J. N. Mourjopoulos, and P. D. Hatziantoniou, "About room response equalization and dereverberation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, USA: IEEE, 2005, pp. 183–186.

[22] Y. Haneda, S. Makino, and Y. Kaneda, "Modeling of a room transfer function using common acoustical poles," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 1992, pp. 213–216.

[23] ——, "Common acoustical pole and zero modeling of room transfer functions," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 320–328, 1994.

[24] Y. Haneda, S. Makino, Y. Kaneda, and N. Koizumi, "ARMA modeling of a room transfer function at low frequencies," English, *J. Audio Eng. Soc. of Japan*, vol. 15, pp. 353–355, Sep. 1994.

[25] M. Karjalainen, P. A. A. Esquef, P. Antsalo, A. Mäkivirta, and V. Välimäki, "Frequency-Zooming ARMA Modeling of Resonant and Reverberant Systems," *J. Audio Eng. Soc. (AES)*, vol. 50, no. 12, pp. 1012–1029, Dec. 2002.

[26] C. Evers, "Blind dereverberation of speech from moving and stationary speakers using sequential Monte Carlo methods," English, PhD thesis, University of Edinburgh, 2010.

[27] S. Furui, *Digital Speech Processing, Synthesis, and Recognition*, eng, ser. Electrical engineering and electronics ; 55. New York, 1989.

[28] K. Kalpakis, D. Gada, and V. Puttagunta, "Distance measures for effective clustering of ARIMA time-series," English, in *Proc. IEEE Intl. Conf. on Data Mining (ICDM)*, San Jose, California, USA, 2001, pp. 273–280.

[29] J. R. Hopgood and P. J. W. Rayner, "Blind single channel deconvolution using nonstationary signal processing," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 476–488, Sep. 2003.

[30] M. Karjalainen, P. Antsalo, A. Mäkivirta, T. Peltonen, and V. Välimäki, "Estimation of modal decay parameters from noisy response measurements," *J. Audio Eng. Soc. (AES)*, vol. 11, pp. 867–878, 2002.

[31] J.-D. Polack, "La transmission de l'énergie sonore dans les salles," PhD thesis, INST_UM, Dec. 1988.

[32] W. C. Sabine, *Collected Papers on Acoustics (Originally 1921)*. 1993.

[33] A. H. Moore, M. Brookes, and P. A. Naylor, "Room identification using roomprints," in *Proc. Audio Eng. Soc. (AES) Conf. on Audio Forensics*, Jun. 2014.

[34] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "Proceeding of the ACE Challenge," INST_ICL, New Paltz, NY, USA, Proceedings, Oct. 2015.

[35] P. P. Parada, D. Sharma, P. A. Naylor, and T. van Waterschoot, "Reverberant speech recognition exploiting clarity index estimation," *EURASIP J. on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–12, 2015.

[36] Y. Dong and D. Li, *Automatic Speech Recognition : A Deep Learning Approach*. Springer-Verlag, 2014.

[37] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep Neural Network Baseline for DCASE Challenge 2016," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[38] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. New York: MacMillan, 1993.

[39] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, ser. Signal Processing Series. New Jersey: Prentice Hall, 1993.

[40] T. Mitchell, *Machine Learning*, ser. McGraw-Hill International Editions. McGraw-Hill, 1997.

[41] G. López, L. Quesada, and L. A. Guerrero, "Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces," in *Advances in Human Factors and Systems Interaction*, I. L. Nunes, Ed., Cham: Springer International Publishing, 2018, pp. 241–250.

[42] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: Learning Lip Sync from Audio," *ACM Trans. Graph.*, vol. 36, no. 4, 95:1–95:13, Jul. 2017.

[43] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943.

[44] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[45] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy Layer-wise Training of Deep Networks," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06, Canada: MIT Press, 2006, pp. 153–160.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012, pp. 1097–1105.

[47] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. 2014.

[48] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 369–376.

[49] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Computing Research Repository*, vol. abs/1409.3215, 2014.

[50] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. C. Courville, "Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks," *Computing Research Repository*, vol. abs/1701.02720, 2017.

[51] Y. Zhang, W. Chan, and N. Jaitly, "Very Deep Convolutional Networks for End-to-End Speech Recognition," *Computing Research Repository*, vol. abs/1610.03022, 2016.

[52] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Budapest, Hungary, 2016, pp. 1128–1132.

[53] S. Mun, S. Park, D. Han, and H. Ko, "Generative Adversarial Network Based Acoustic Scene Training Set Augmentation and Selection Using SVM Hyper-Plane," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[54] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, p. 947, Jun. 2000.

[55] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, "On rectified linear units for speech processing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3517–3521.

[56] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudik, Eds., ser. Proceedings of Machine Learning Research, vol. 15, Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323.

[57] S. Theodoridis, *Machine Learning*. Academic Press, 2015.

[58] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Neurocomputing: Foundations of Research," in, J. A. Anderson and E. Rosenfeld, Eds., Cambridge, MA, USA: MIT Press, 1988, pp. 696–699.

[59] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Computing Research Repository*, vol. abs/1412.6980, 2014.

[60] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[61] L. Zhang and P. N. Suganthan, "Visual Tracking with Convolutional Neural Network," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2015, pp. 2072–2077.

[62] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computing Research Repository*, vol. abs/1409.1556, 2014.

[63] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. 2016.

[64] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[65] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[66] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 4580–4584.

[67] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Computing Research Repository*, vol. abs/1404.7828, 2014.

[68] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *ArXiv e-prints*, Jun. 2014.

[69] D. P Kingma and M. Welling, "Auto-Encoding Variational Bayes," *ArXiv e-prints*, Dec. 2013.

[70] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *ArXiv e-prints*, Jan. 2017.

[71] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks," *ArXiv e-prints*, Jan. 2017.

[72] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *Computing Research Repository*, vol. abs/1606.03498, 2016.

[73] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY: Springer-Verlag, 2009.

[74] G. Forman and I. Cohen, "Learning from Little: Comparison of Classifiers Given Little Training," in *Knowledge Discovery in Databases: PKDD 2004*, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 161–172.

[75] MATLAB, *Version 9.2.0.556344 (R2017a)*. Natick, Massachusetts, 2017.

[76] L. Breiman, J. H. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. CRC Press, 1984.

[77] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[78] K. Q. Weinberger and L. K. Saul, "Fast Solvers and Efficient Implementations for Distance Metric Learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, Helsinki, Finland: ACM, 2008, pp. 1160–1167.

[79] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[80] Z. Wang and X. Xue, "Multi-Class Support Vector Machine," in *Support Vector Machines Applications*, Y. Ma and G. Guo, Eds., Cham: Springer International Publishing, 2014, pp. 23–48.

[81] J. Fürnkranz, "Round Robin Classification," *J. Mach. Learn. Res.*, vol. 2, pp. 721–747, Mar. 2002.

[82] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, "Introduction," in *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–23.

[83] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.

[84] J Handl and J Knowles, "An Evolutionary Approach to Multiobjective Clustering," English, *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, Feb. 2007.

[85] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[86] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *CISTM*, vol. 3, no. 1, pp. 1–27, 1974.

[87] C. Evers and J. R. Hopgood, "Parametric modelling for single-channel blind dereverberation of speech from a moving speaker," *IET Signal Processing*, vol. 2, no. 2, pp. 59–74, Jun. 2008.

[88] B. Kapralos, N. Mekuz, A. Kopinska, and S. Khattak, "Dimensionality reduced HRTFs: A comparative study," in *Proc. ACM Intl. Conf. on Advances in Computer Entertainment Technology (ACE)*, Yokohama, Japan: ACM, Dec. 2008, pp. 59–62.

[89] R. Duraiswami and V. C. Raykar, "The Manifolds of Spatial Hearing," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, Pennsylvania, USA, Mar. 2005, pp. 285–288.

[90] E. Lehmann and A. Johansson, "Diffuse reverberation model for efficient image-source simulation of room impulse responses," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1429–1439, Aug. 2010.

[91] K. Meesawat and D. Hammershoi, "The Time When the Reverberation Tail in a Binaural Room Impulse Response Begins," in *Proc. Audio Eng. Soc. (AES) Convention*, New York, NY, USA, Oct. 2003.

[92] V. Valimaki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 5, pp. 1421–1448, Jul. 2012.

[93] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proc. National Academy of Sciences*, 2013.

[94] L. Remaggi, P. J. B. Jackson, P. Coleman, and W. Wang, "Acoustic Reflector Localization: Novel Image Source Reversion and Direct Localization Methods," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 2, pp. 296–309, Feb. 2017.

[95] F. Antonacci, A. Sart, and S. Tubaro, "Geometric reconstruction of the environment from its response to multiple acoustic emissions," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 2822–2825.

[96] I. Dokmanic, Y. Lu, and M. Vetterli, "Can one hear the shape of a room: The 2-D polygonal case," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 321–324.

[97] S. Tervo and T. Tossavainen, "3D room geometry estimation from measured impulse responses," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[98] A. H. Moore, M. Brookes, and P. A. Naylor, "Room geometry estimation from a single channel acoustic impulse response," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Marrakech, Morocco, Sep. 2013, pp. 1–5.

[99] G. P. Nava, Y. Yasuda, Y. Sato, and S. Sakamoto, "In-situ estimation of acoustic impedance on the surfaces of a room for inverse sound rendering," in *Proc. Int. Conf. on Noise and Vibration Engineering (ISMA)*, Lueven, Belgium, 2006.

[100] D. Arteaga, D. Garcia-Garzón, T. Mateos, and J. Usher, "Scene Inference from Audio," Rome, Italy, May 2013.

[101] S. Pelzer and M. Vorländer, "Inversion of a room acoustics model for the determination of acoustical surface properties in enclosed spaces," *Proceedings of Meetings on Acoustics*, vol. 19, no. 1, p. 015 115, 2013.

[102] A. Pilch, "Optimization in the validation of the room acoustic model," in *Proceedings of EuroRegio*, Porto, Portugal, 2016.

[103] K. Saksela, J. Botts, and L. Savioja, "Optimization of absorption placement using geometrical acoustic models and least squares," *The Journal of the Acoustical Society of America*, vol. 137, no. 4, EL274–EL280, 2015.

[104] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material Recognition in the Wild With the Materials in Context Database," in *Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, Jun. 2015.

[105] C. Schissler, C. Loftin, and D. Manocha, "Acoustic Classification and Optimization for Multi-Modal Rendering of Real-World Scenes," *IEEE Trans. on Visualization and Computer Graphics*, vol. 24, no. 3, pp. 1246–1259, Mar. 2018.

[106] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

[107] S. Adavanne and T. Virtanen, "Sound Event Detection Using Weakly Labeled Dataset with Stacked Convolutional and Recurrent Neural Network," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[108] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[109] Q. Kong, Y. Xu, W. Wang, and M. Plumbley, "Audio set classification with attention model: A probabilistic perspective," in *ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ser. IEEE ICASSP 2018 Proceedings, IEEE, Sep. 2018.

[110] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 Challenge Setup: Tasks, Datasets and Baseline System," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, Nov. 2017, pp. 85–92.

[111] L. Cuccovillo, S. Mann, M. Tagliasacchi, and P. Aichroth, "Audio tampering detection via microphone classification," in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2013, pp. 177–182.

[112] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[113] T Virtanen, M. Plumbley, and D Ellis, "Introduction to sound scene and event analysis," in *Computational Analysis of Sound Scenes and Events*, T Virtanen, M. Plumbley, and D Ellis, Eds., Cham, Switzerland: Springer, 2018, pp. 3–12.

[114] J. Mourjopoulos, A. Tsopanoglou, and N. Fakotakis, "A vector quantization approach for room transfer function classification," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, 1991, pp. 3593–3596.

[115] S. Bharitkar and C. Kyriakakis, "Perceptual multiple location equalization with clustering," in *Signals, Systems and Computers, 2002. Conf. Record of the Thirty-Sixth Asilomar Conf. On*, vol. 1, IEEE, 2002, pp. 179–183.

[116] I. Omiciuolo, A. Carini, and G. L. Sicuranza, "Multiple position room response equalization with frequency domain fuzzy c-means prototype design," in *Proc. Intl. Workshop on Acoustic Signal Enhancement (IWAENC)*, 2008.

[117] N. Peters, H. Lei, and G. Friedland, "Name that room: Room identification using acoustic features in a recording," in *Proceedings of the 20th ACM International Conference on Multimedia*, 2012, pp. 841–844.

[118] A. H. Moore, P. A. Naylor, and M. Brookes, "Room Identification Using Frequency Dependence of Spectral Decay Statistics," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 6902–6906.

[119] H. Malik and H. Mahmood, "Acoustic environment identification using unsupervised learning," *Security Informatics*, vol. 3, no. 1, p. 1, 2014.

[120] A. H. Moore, M. Brookes, and P. A. Naylor, "Roomprints for forensic audio," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2013.

[121] J. N. Mourjopoulos, "Digital equalization of room acoustics," *J. Audio Eng. Soc. (AES)*, vol. 42, no. 11, pp. 884–900, Nov. 1994.

[122] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "Estimation of room acoustic parameters: The ACE challenge," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 10, pp. 1681–1693, Oct. 2016.

[123] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[124] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[125] D. Steinberg, *CART: Classification and Regression Trees*. CRC Press, 2009, vol. 9.

[126] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," in *Learning from Data*, PUB-SV, 1996, pp. 199–206.

[127] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus," INST_LDC, Philadelphia, Corpus, 1993.

[128] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[129] T. J. Cox, F. Li, and P. Darlington, "Extracting room reverberation time from speech using artificial neural networks," *J. Audio Eng. Soc. (AES)*, vol. 49, no. 4, pp. 219–230, 2001.

[130] E Benetos, D Stowell, and M. Plumbley, "Approaches to complex sound scene analysis," in *Computational Analysis of Sound Scenes and Events*, T Virtanen, M. Plumbley, and D Ellis, Eds., Cham, Switzerland: Springer International Publishing, 2018, pp. 215–242.

[131] Y. Han and J. Park, "Convolutional Neural Networks with Binaural Representations and Background Subtraction for Acoustic Scene Classification," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[132] Z. Ren, Q. Kong, K. Qian, M. D. Plumbley, and B. W. Schuller, "Attention-based Convolutional Neural Networks for Acoustic Scene Classification," in *3rd Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2018 Workshop)*, ser. DCASE 2018 Workshop Proceedings, 2018.

[133] A. Graves, N. Jaitly, and A. r Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2013, pp. 273–278.

[134] T. Sainath, R. J. Weiss, K. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel Signal Processing with Deep Neural Networks for Automatic Speech Recognition," *IEEE /ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 965 –979, 2017.

[135] Y.-T. Zhou, R. Chellappa, A. Vaid, and B. K. Jenkins, "Image restoration using a neural network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1141–1151, Jul. 1988.

[136] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *Computing Research Repository*, vol. abs/1207.0580, 2012.

[137] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[138] S. Ruder, "An overview of gradient descent optimization algorithms," *Computing Research Repository*, vol. abs/1609.04747, 2016.

[139] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[140] Y. Bengio, *Neural Networks: Tricks of the Trade*. Springer, 2012.

[141] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[142] D. Ba, F. Ribeiro, C. Zhang, and D. Florencio, "L1 regularized room modeling with compact microphone arrays," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, USA, Mar. 2010, pp. 157–160.

[143] F. Antonacci, J. Filos, M. R. P. Thomas, E. A. P. Habets, A. Sarti, P. A. Naylor, and S. Tubaro, "Inference of room geometry from acoustic impulse responses," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 10, pp. 2683–2695, Dec. 2012.

[144] I. Kelly and F. Boland, "Detecting arrivals in room impulse responses with dynamic time warping," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 7, pp. 1139–1147, Jul. 2014.

[145] G Defrance, L Daudet, and J. D. Polack, "Using matching pursuit for estimating mixing time within room impulse responses," *Acta Acustica united with Acustica*, vol. 95, no. 6, pp. 1071–1081, 2009.

[146] C. H. Jeong, J. Brunskog, and F. Jacobsen, "Room acoustic transition time based on reflection overlap," English, *J. Acoust. Soc. Am.*, vol. 127, pp. 2733–2736, 2010.

[147] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti, "Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, Jul. 2007.

[148] D. A. Bohn, "Environmental Effects on the Speed of Sound," in *Audio Engineering Society Convention 83*, Oct. 1987.

[149] "EM32 Eigenmike microphone array release notes (v17.0)," INST_MHA, NJ USA, Hardware, Oct. 2013.

[150] K. A. Kosanovich, M. J. Piovoso, K. S. Dahl, J. F. MacGregor, and P. Nomikos, "Multi-way PCA applied to an industrial batch process," in *Proceedings of 1994 American Control Conference - ACC '94*, vol. 2, Jun. 1994, 1294–1298 vol.2.

[151] F. E. Toole, "Loudspeaker Measurements and Their Relationship to Listener Preferences: Part 1," *J. Audio Eng. Soc*, vol. 34, no. 4, pp. 227–235, 1986.

[152] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale -Regularized Least Squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[153] W. Forst and D. Hoffmann, *Optimization—Theory and Practice*. Springer-Verlag, 2010.

[154] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, Apr. 1979.

[155] A. Wabnitz, N. Epain, C. Jin, and A. van Schaik, "Room acoustics simulation for multichannel microphone arrays," in *Proceedings of the Intl. Symp. on Room Acoustics*, 2010.

[156] *ISO 354:2003: Acoustics — Measurement of sound absorption in a reverberation room.*

[157] *ISO 10534-2: Determination of sound absorption coefficient and acoustic impedance with the interferometer.*

[158] H. Lim, J. Park, and Y. Han, "Rare Sound Event Detection Using 1D Convolutional Recurrent Neural Networks," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[159] E. Cakir and T. Virtanen, "Convolutional Recurrent Neural Networks for Rare Sound Event Detection," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[160] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *Computing Research Repository*, vol. abs/1412.3555, 2014.

[161] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An Empirical Exploration of Recurrent Network Architectures," in *Proc. of the Int. Conf. on on Machine Learning*, ser. ICML'15, Lille, France: JMLR.org, 2015, pp. 2342–2350.

[162] E. Castillo, A. J. Conejo, P. Pedregal, R. Garcia, and N. Alguacil, "Mixed-Integer Linear Programming," in *Building and Solving Mathematical Programming Models in Engineering and Science*, Wiley-Blackwell, 2011, pp. 161–182.

[163] L. Breiman and P. Spector, "Submodel Selection and Evaluation in Regression. The X-Random Case," *International Statistical Review / Revue Internationale de Statistique*, vol. 60, no. 3, pp. 291–319, 1992.

[164] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *Computing Research Repository*, vol. abs/1603.04467, 2016.

[165] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[166] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proc. of the 23rd International Conf. on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 233–240.

[167] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA, 2008.

[168] D. R. Morgan, J. Benesty, and M. Sondhi, "On the evaluation of estimated impulse responses," *IEEE Signal Process. Lett.*, vol. 5, no. 7, pp. 174–176, Jul. 1998.

[169] A. H. Gray Jr. and J. D. Markel, "Distance measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 5, pp. 380–391, Oct. 1976.

[170] C. Evers and J. Hopgood, "Multichannel online blind speech dereverberation with marginalization of static observation parameters in a rao-blackwellized particle filter," English, *J. of Signal Processing Systems*, vol. 63, no. 3, pp. 315–332, 2011.

[171] T. W. Parks and C. S. Burrus, *Digital Filter Design*. Wiley, 1987.

[172] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," *Computing Research Repository*, vol. abs/1608.04363, 2016.

[173] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Detection," *Computing Research Repository*, vol. abs/1604.07160, 2016.

[174] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings," *Computing Research Repository*, vol. abs/1604.00861, 2016.

[175] J. Schlüter and T. Grill, "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks.," in *Intern. Soc. for Music Information Retrieval Conf. (ISMIR)*, Malaga, Spain, Oct. 2015, pp. 121–126.

[176] A. Sriram, H. Jun, Y. Gaur, and S. Satheesh, "Robust Speech Recognition Using Generative Adversarial Networks," *Computing Research Repository*, vol. abs/1711.01567, 2017.

[177] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition," *Computing Research Repository*, vol. abs/1711.05747, 2017.

[178] C. Li, T. Wang, S. Xu, and B. Xu, "Single-channel Speech Dereverberation via Generative Adversarial Training," *ArXiv e-prints*, Jun. 2018.

[179] K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, and L. Xie, "Investigating Generative Adversarial Networks based Speech Dereverberation for Robust Speech Recognition," *Computing Research Repository*, vol. abs/1803.10132, 2018.

[180] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *Computing Research Repository*, vol. abs/1411.1784, 2014.

[181] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation," *Computing Research Repository*, vol. abs/1704.02510, 2017.

[182] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Computing Research Repository*, vol. abs/1502.03167, 2015.

[183] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep Cross-Modal Audio-Visual Generation," *Computing Research Repository*, vol. abs/1704.08292, 2017.

[184] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. N. Metaxas, "Stack-GAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks," *Computing Research Repository*, vol. abs/1612.03242, 2016.