Machine-learning-based estimation of room acoustic parameters

Ricardo Falcón Pérez

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 26.11.2018

Thesis supervisor:

Prof. Ville Pulkki

Thesis advisor:

MSc. Leo McCormack

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Ricardo Falcón Pérez

Title: Machine-learning-based estimation of room acoustic parameters

Date: 26.11.2018 Language: English Number of pages: 6+72

Department of Signal Processing and Acoustics

Professorship: Acoustics and Audio Signal Processing (S-89)

Supervisor: Prof. Ville Pulkki

Advisor: MSc. Leo McCormack

Traditional methods to study sound propagation inside rooms can be divided in two approaches: geometrical models and wave-based models. In the former, sound is analyzed as rays, giving a valid approximation for high frequencies while failing to model certain wave effects such as diffraction or inference. The latter, finds solutions for the wave equation, providing better accuracy at the cost of much higher computational complexity.

This thesis presents a proof of concept for a novel machine learning method to estimate a set of typical room acoustics parameters using only geometrical information as input features. First, a room acoustics dataset composed of real world acoustical measurements is analyzed and processed using microphone array encoding techniques to extract room impulse responses and acoustical absorption area for multiple directions. The dataset is explored to identify correlation between features and general properties, including a low dimensionality representation for visualization.

The proposed method uses geometrical features as input for a neural network model that estimates room acoustics parameters, such as reverberation time (T60), and early decay time (EDT). For reverberation time, this model is evaluated against the Sabine method and the results show much higher accuracy, especially at low frequencies. The method is then expanded to include input features for the locations of the source and microphone, where the results also achieve high performance.

Furthermore, an hyperparameter optimization procedure using random search reveals three main findings. First, that a large range of neural networks architectures, even with very few trainable parameters, achieve high performance. Second, the depth of the models has little influence on the results. Third, the benefit of increasing the amount of training data examples for a single loudspeaker saturates after around 100 examples.

Keywords: Room acoustics, spatial audio, room impulse response, microphone array, machine learning, neural networks, data analysis

Preface

This thesis work was conducted at the Department of Signal Processing and Acoustics in Aalto Univeristy in Finland, as part of the degree requirement for the master program in Acoustics and Audio Signal Processing.

Firstly, I would like to thank my supervisor Prof. Ville Pulkki for allowing me to contribute to his ambitious project and providing me with guidance and material support. Special thanks is extended to Leo McCormack for showing me the light of spatial audio and spherical harmonics; and to Luiza Sayfullina, for her vast recommendations on machine learning and deep learning models. I am also grateful to everyone in the Aku Lab for their help, advice, and foosball matches. More special thanks are sent to Georg Götz for dedicating his summer to taking all those measurements.

It is of the utmost importance to mention that the majority of the technical work done for this project would not have been possible without careful exploitation of the Ballmer's peak. For long considered a myth, recent developments support what programmers discovered many years ago; certain beverages increase problem solving skills significantly, and enable creativity [1]. More than once, whenever I felt trapped in a labyrinth inside one of few thousands of lines of code of the project, a cold brew (along with the appropriate IDM or nu-jazz soundtrack) were there to show the solution, or a crafty workaround. It is well know that barley, and the miraculous liquid refreshment know as beer, were crucial to the establishment of civilization (the so called beer-before-bread theory [2]), but I believe that anyone working with software can now vouch that beer is also vital to mankind's continuous development, and will be a key ingredient to reach the technological singularity. I am very thankful to the large variety and high quality of craft brews available in Finland.

Otaniemi, 26.11.2018

Ricardo Falcón Pérez

Contents

Symbols and abbreviations

Symbols

c	speed of sound
f_s	sampling frequency

Operators

$\nabla f(x)$	gradient of function $f(x)$
$\frac{\mathrm{d}}{\mathrm{d}t}$	derivative with respect to variable t
$\frac{\partial}{\partial t}$	partial derivative with respect to variable t
$\sum_i$	sum over index i
$\mathbf{A} \cdot \mathbf{B}$	dot product of vectors $\mathbf{A}$ and $\mathbf{B}$

Abbreviations

EDT	Early Decay Time
IR	Impulse Response
RIR	Room Impulse Response
SPL	Sound Pressure Level
T_{60}	Reverberation time
EDT	Early Decay Time
C_{50}	Clarity Index at 50 ms
DRR	Direct to Reverberant Ratio
L	Sound Level
NN	Neural Network
CNN	Convolutional Neural Network
MLP	Multi Layer Perceptron
FC	Fully Connected
SVM	Support Vector Machine
PCA	Principal Component Analysis
ML	Machine Learning
AI	Artificial Intelligence
SGD	Stochastic Gradient Descent
KNN	K-nearest neighbors
MSE	Mean Square Error
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

Chapter 1

Introduction

1.1 Motivation

Sound propagates across the air as waves. Indoors, these waves travel uncontested until they reach an obstacle (e.g. surfaces delimiting the space itself or objects inside the space). The original sound wave will interact with every one of these obstacles, sometimes in complex ways. From the point of view of a listener located inside the space, for example, an audience member inside a concert hall listening to an orchestra; the quality of the auditory experience will depend not only on the performance of the orchestra, but also on the inherent acoustical properties of the hall, which modify and alter the perceived sound.

Therefore, the study and analysis of how sound is manipulated inside rooms is important, as it explains why some venues enhance a listening experience while others degrade it. In other words, by understanding the physics of sound propagation, it is possible to, for instance, embellish the music played in a performance venue, or to optimize the intelligibility of speech in a classroom.

Sound propagation is more or less well understood: it travels as mechanical waves [3]. However, even if the theory behind this phenomenon is simple, its modeling is still very computationally costly, which limits either the accuracy or the availability of results. Traditionally, there are two main approaches to model sound propagation. The first approach considers sound as rays. Although this approximation is valid for high frequencies (i.e. when the wavelength is small compared to the dimensions of the obstacles), this approach fails to model wave effects such as diffraction or interference [4].

The second approach models sound propagation as waves, and solves the corresponding wave equation (with multiple methods for both time and frequency domains). Here the accuracy is much higher than geometrical acoustics, but at the expense of increased computational demands [5], [6].

However, recent advances in Artificial Intelligence —and machine learning in particular —make a third approach now possible [7] [8]. With enough data, a mathematical model can be trained to emulate all the transformations that any given input sound may be subjected to under a certain room conditions. This process can be accounted for without the need to simulate or solve complex equations, as it is

based purely on *imitation*, whereby similar input signals will always produce similar output sounds.

Although full acoustical modelling based on data is quite an ambitious goal, a more simple case is to start by predicting not the physical propagation itself, but a set of specific acoustical parameters that are measurable in practice. For example, the reverberation time T_{60} , which is perhaps the most basic parameter to describe how the sound experience inside a room might be, can be modeled as a regression task. Here, the objective is to predict values for T_{60} given information inside the room or the signal itself. In [9], such a system was developed, in which noisy recordings inside a room were used to train a model and predict the T_{60} values. A different use case could be to predict the same values based only on geometrical information or even pictures.

The main focus of this thesis is the prediction of a full set of room acoustics parameters using data such : room geometry, acoustical absorption properties, or spherical photographs. To do so, we present the analysis of a dataset composed of real life acoustical measurements, explore some of its properties, and then build, train and evaluate machine learning models based on this data. This project is a proof of concept, that verifies the validity of a data driven approach for acoustic modelling.

1.2 Outline

Chapter 2 begins with an introduction to room acoustics, including some terminology and definitions for the acoustical parameters which are later used as estimation targets. It is followed by a brief description of spatial audio and how microphone array encoding is used to process the recorded signals. The last part of the chapter has an overview of general machine learning topics, including some terminology, problem definitions, neural networks essentials, and a short walkthrough over typical pitfalls and useful tricks that need to be considered in a regular machine learning problem. Readers with knowledge of acoustics and spatial audio can skip directly to section 2.3, while machine learning experts should start with sections 2.1 and 2.2 and then continue with the following chapters.

In chapter 3, the data used for the thesis is described and analyzed. First there is a cursory review of the few public datasets for room acoustics that are available. Then the raw data is characterized, along with the process used to extract the room acoustics parameters and a few useful features. There is also an exploratory data analysis that illustrates some of the properties of the data.

In chapter 4 the methodology used for all the experiments in this thesis is detailed. It starts enumerating the steps for the data preprocessing, where the data is prepared for the machine models. It is followed by a definition of the performance metrics used to evaluate the models, including MSE, MAE, MAPE and accuracy. Lastly, the actual machine learning models are specified, including characteristics such as neural networks architecture, input and output layers, and the hyperparameter optimization procedure employed.

Chapter 5 shows the results from the experiments realized. First it compares the proposed method to the Sabine formula, which is used to validate the value of our model. Afterwards, an unsupervised analysis of the data characterizes the data further. Finally, the prediction results for the models are listed, including insights on how some hyperparameters influence the performance.

In chapter 6 the results presented earlier are discussed, where some interesting findings are noted. Moreover, some thoughts are shared for future research. At last, chapter 7 summarizes all the findings in this thesis.

Chapter 2

Background

This chapter includes basic concepts and theoretical knowledge required to understand this work. First there is an introduction to room acoustics and acoustical measurements in general, followed by a brief explanation of spatial audio and topics related to spherical microphone array processing. Finally, there is a summary of basic elements of machine learning, neural networks and how to set up a machine learning project.

2.1 Room Acoustics

In general, for an ideal omnidirectional source, sound waves will propagate evenly in all directions. Moreover, sound waves interact with obstacles and objects in various ways. The sound experienced inside a room is the combination of the direct sound (which travels straight from the source to the listener) and the sound that bounces and reflects back from the surfaces inside the room. The main sources of reflections are the architectural elements of the room, such as the walls, floor and ceiling, but anything some geometrical shape inside the room will affect sound propagation, be it furniture, or even people. What happens to sound as it hits a surface depends on the materials of the surface and the angle of incidence. Largely, sound can either pass through an object, be absorbed by it, or reflect back (in specular or difrracted way) [3] [10]. In practice, absorption and reflection both happen at the same time, at different frequencies, and angles.

There are many different types of rooms, which can have different size, shape, furniture or construction materials. Architectural elements of the room are more or less fixed after construction, while the elements inside the room are easily changeable. In the end, the configuration of a room will determine sound propagation and the listening experience inside it. Furthermore, different activities have different requirements. For example, a classroom will prioritize speech intelligibility, while a concert hall should embellish the experience of musical performances.

2.1.1 Absorption

Sound absorption occurs when sound energy collides with a surface and is transformed into another type of energy, typically heat. How much energy is transformed depends on the properties of the surface materials, where usually porous materials are used for sound absorbers. The sound absorption coefficient α measures the fraction of sound energy that is absorbed by a surface, where $\alpha = 1$ corresponds to full absorption, and $\alpha = 0$ describes a fully reflective surface. That said, it is common for materials to have different α values for different frequency bands, so these values are commonly reported by octave bands from 125 to 4000 Hz. The unit for α is called Sabin.

To compute the total absorption in room, it is necessary to consider the different absorption coefficients and the surfaces they cover. This can be calculated as

$$A = \sum_{i=1}^{n} \alpha_i S_i = \alpha_1 S_1 + \alpha_2 S_2 + ... + \alpha_n S_n, \tag{2.1}$$

where A is the total absorption area, S_1, S_2, etc is the area of surface and α_i is the absorption coefficient that corresponds to the ith surface [3] [10].

It is important to note that some materials report absorption coefficients values larger than 1. In principle, this should not be possible, as $\alpha = 1$ corresponds to 100% absorption. However, these values can occur due to the test procedure used in laboratory measurements. In short, the absorption provided by the edges of a test sample are not considered when determining the surface magnitude [11].

2.1.2 Reflection and Diffusion

If a surface does not absorb the sound, it bounces back (or goes through). If the surface curve/shape is much larger than the wavelength, then the sound is reflected at the same angle as the incidence, compared to the surface normal. This reflection behaves much like a mirror and is called specular reflection. These reflections can be modeled as rays or image sources [4]. If the surface is rough, compared to the wavelength, the sound will be reflected in multiple directions, as the product of scattered reflection or diffusion [3].

2.1.3 Diffraction

Diffraction occurs whenever a sound wave encounters an edge, where the direction of propagation changes. This property allows, for example, to hear sound around corners in a room. The sound wave bends around the edge, and the amount of bending depends on the size of the edge compared to the wavelength, such that long wavelengths bend more than shorter ones [3].

2.1.4 Room impulse response

Sound traveling inside a room will be affected by the phenomena mentioned earlier. Therefore, the signal captured by a microphone in an arbitrary listening position

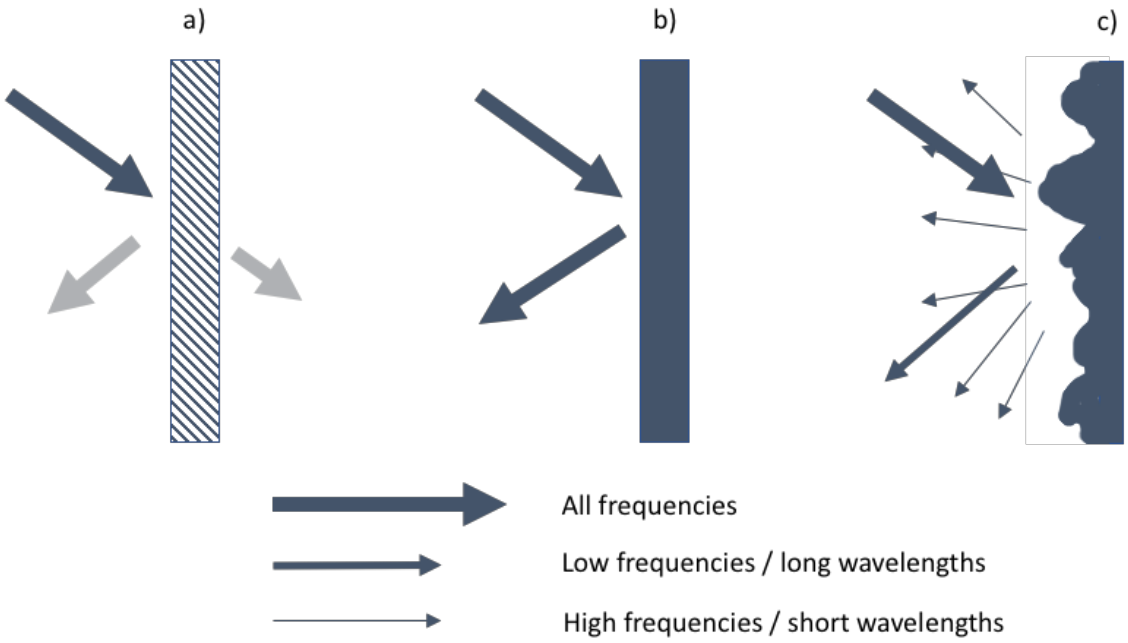


Figure 2.1: Basic concept of sound absorption, reflection and diffusion. Sound arriving at an absorptive barrier (a) will be absorbed, however, depending on the properties of the material, a fraction of the sound will be reflected and another fraction will pass through the barrier. If the surface is rigid and flat compared to the wavelength of the incident sound (b), the sound will be reflected back. However, if the accidents of the surface are large compared to the wavelength of the incident sound (c), sound will be diffused, where reflection will happen in many directions at the same time. Adapted from [12].

inside the room will be different than the sound emitted by a source at some other position. All the modifications suffered by the signal can be summarized by the impulse response if the system is linear and time invariant (LTI) [3] [13].

The impulse response can be split into 3 parts: 1) direct sound, 2) early reflections, and 3) reverberation. The direct part corresponds to the first few milliseconds and describes how the signal generated by the source arrives at the listener. In ideal conditions, this direct component is the same as the transmitted signal, slightly delayed due to the travel time. The early reflections are the result of the direct sound bouncing back after hitting obstacles and arriving at the listening position shortly after the direct sound. These reflections only hit a small number of boundaries before arriving at the listening position. Finally, the reverberation occurs after many reflections. Figure 2.2 shows an illustration of a room impulse response and the main sections.

The room impulse response contains a large amount of information, corresponding to changes in time, level, spectrum and direction of incidence (for multichannel impulses). Moreover, the impulse response takes into account the full signal path, including the characteristics of the source (such as loudspeakers), the room itself, and the measuring device or microphone. In practice, not all this information is

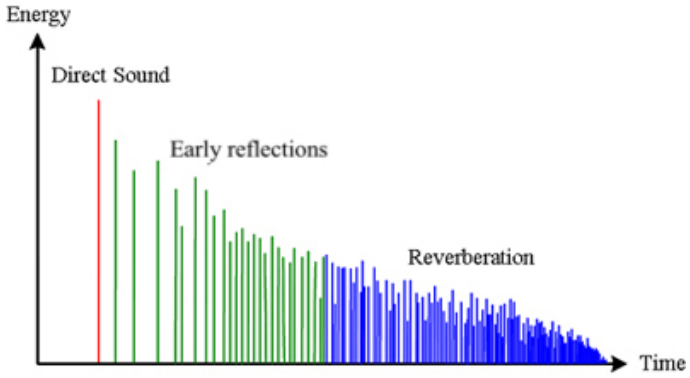


Figure 2.2: Theoretical room impulse response showing the 3 sections. Adapted from [14].

needed to estimate how sound is perceived inside the room [3]. Therefore, several parameters or statistics are extracted and computed from the raw impulse response. The most relevant of these parameters are explained in the following section.

2.1.5 Reverberation Time (T_{60})

Reverberance is one of the most important parameters that characterize a room [3]. It is the product of all the reflections occurring inside a room and can greatly affect the listening experience. Long reverberation can affect speech intelligibility because the short consonant sounds get drowned by long, vocal sounds. On the other hand, music becomes dull and loses power if the reverberation is too short.

The reverberation time T_{60} is the time period in which a sound is attenuated enough to not be listenable anymore. Technically, this happens when the sound level has dropped 60 dB after the sound generation process has stopped. It can be determined empirically, by measuring the sound pressure level and noting the time it takes to drop. It can also be extracted from the impulse response using one of the methods denoted in technical standards [15].

Most calculations for T_{60} assume certain conditions that might hold for small rooms, such as the presence of a perfect diffuse field. Given that standing waves and strong specular reflections can dominate inside small rooms, the results from estimations need to be interpreted with care.

Additionally, T_{60} can be approximated using the Sabine formula [16]

$$T_{60} = \frac{24 \times \ln(10) V}{cA}, \tag{2.2}$$

where V is the volume of the room [m^3], c is the speed of sound in [m/s], and A is the total absorption area in the room [sabine / m^2], computed as described in equation 2.1. The unit for T_{60} is seconds, and typical values for small rooms are in 0.3 - 1 s, whereas larger rooms can go up to 2 s or longer.

2.1.6 Early Decay Time (EDT)

The early decay time (EDT) is similar to T_{60}, in that it measures reverberation. The difference is that EDT considers the amount of time it takes for sound to drop from 0 dB to -10 dB, scaled so that it corresponds to the decay from 0 dB to -60 dB [17]. Therefore, it measures the time required to drop the sound level by 10 dB after the sound source has stopped, rather than 60 dB.

Because EDT focuses on the early part of reverberation and not the tail, it describes early reflections better [3]. Moreover, EDT can be a more appropriate approximation to the perceived reverberation in the room if the whole reverberation response is uneven [3].

2.1.7 Sound Level (L)

In this work, the sound level L is the maximum value of the energy decay curve (EDC). This parameter is not commonly measured to describe room acoustics, but here it is used to compare the relative level of the direct sound of multiple impulse responses for the same event. In practice, this correlates with the direction of arrival.

The EDC is computed as

$$\mathrm{EDC}(t) = \int_t^\infty h^2(t)dt, \tag{2.3}$$

where h is the impulse response [3],[18]. The unit for L is dB.

2.1.8 Clarity (C_{50})

The clarity parameter (C_{50}) describes how easy it is to distinguish individual sounds, instead of a blurred combination of different events. The theory behind this parameter is that reflections arriving within a window of no more than 50-80 ms after the direct sound will be perceived as a single event, due to temporal integration. These reflections will increase the perceived amplitude of the sound (and in some cases modify the timbre).

C_{50} is measured by calculating the ratio between the energy of the early reflections and the late responses. The definition for early response can be set at 50 or 80 ms depending on the application. For this work, 50 ms is used. It is computing as

$$C_{50} = 10log_{10}\frac{\int_0^{50_{ms}} p^2(t)dt}{\int_{50_{ms}}^\infty p^2(t)dt}, \tag{2.4}$$

where $p(t)$ is the sound pressure level [17]. The unit for C_{50} is dB, where positive values corresponds to a strong early response, while negative values follows a strong late response.

2.1.9 Direct to Reverberant Ratio (DRR)

The direct to reverberant ratio (DRR) is very similar to the C_{50}, as it also measures the ratio between the energy of the early and late responses. However, here the window for the early part is set to 2.5 ms. Consequently, this parameter focuses only on the impulsive part of the direct sound and does not consider early reflections. This can be important for percussive sounds such as clicks, were even reflections arriving at 30 ms can be perceived as distinct events if the level is high enough [19], [20]. DRR is computed as

$$DRR = 10log_{10}\frac{\int_0^{2.5_{ms}} p^2(t)dt}{\int_{2.5_{ms}}^{\infty} p^2(t)dt}, \tag{2.5}$$

where $p(t)$ is the sound pressure level. The unit for C_{50} is dB, where positive values corresponds to a strong impulsive response, while negative values follows a strong late response.

2.1.10 Filtering by octave bands

The acoustical response of a room depends on the frequency content of the signal. For example, high frequencies are usually easily absorbed by most obstacles or even air, while low frequency components are more difficult to control and suffer from additional problems such as room modes. Therefore, most of the acoustical parameters described in this chapter are usually computed at different frequency bands. To do this, the recorded impulse response is first filtered by octave wide band pass filters and then the aforementioned parameters are computed. On this work, all parameters except C_{50} and DRR are extracted using this process.

2.2 Spatial Audio

The acoustical parameters discussed in the previous section consider the monophonic case, where there are one or more sources, but only one receiver. However, certain properties of the full listening experience can only be captured by having multiple receivers. For example, direction of arrival (DoA) of sound sources and early reflections may only be established if the sound scene is captured by multiple sensors [21], [22]. For room acoustics, the difference of some acoustical parameters between sensors can be used to identify unique positions inside the room. The study of audio across multiple directions is called **spatial audio**.

To enable spatial analysis, multiple sensors are arranged in an array, where the specific configuration depends on the application. For audio, compact spherical arrays with phase-matched microphones are especially popular, due to their portability, and rotational symmetry [23]. To extract spatial information, the signals recorded by each individual microphone need to be processed, using microphone array encoding techniques. Compact spherical arrays are also well suited for analytical spatial encoding of the microphone array signals into an array independent domain; via

the spherical harmonic transform (SHT) [24]. This transform essentially abstracts away the microphone array specifications and allows the ability to generate static virtual microphones pointing at any arbritary direction, using broadband steering vectors [25]; with some commercially available spherical arrays providing reasonable performance at a wide enough frequency range [26], which are suitable for the analysis performed later in this thesis.

2.2.1 Microphone Array Encoding

The microphone array signals are first encoded, by transforming them into the spherical harmonics domain. Spherical harmonics are orthonormal basis functions over the unit sphere, defined as

$$Y_n^m(\theta,\phi) \equiv \sqrt{\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}} P_n^m(sin\theta)e^{im\phi}, \tag{2.6}$$

where $n \in [0,1,2,\ldots,N]$ is the function order, $m \in [-n,n]$ is the function degree, $P_n^m(\cdot)$ are the orthonormal Legendre polynomials, $\theta \in [-\pi/2,\pi/2]$ denotes the elevation angle, and $\phi \in [-\pi,\pi]$ is the azimuth angle. The spherical harmonics functions can be visualized as a color map over the sphere, or as balloon plots where the magnitude shows the distance to the origin and the colors represent the sign of the values. An example of these visualizations is shown in figure 2.3. Real spherical harmonics up to third order are shown in figure 2.4.

A typical spherical microphone array has Q sensors, located at positions $\Omega_q = (\theta,\phi,r)$. The encoding of the microphone signals is estimated up to the maximum number of expansion N as

$$\mathbf{s}(t,f) = \mathbf{W}(f)\mathbf{x}(t,f), \tag{2.7}$$

where $\mathbf{s} \in \mathbb{C}^{(N+1)^2\times 1}$ are the spherical harmonic signals, $\mathbf{x} \in \mathbb{C}^{Q\times 1}$ are the microphone signals in time-frequency domain [1], and $\mathbf{W} \in \mathbb{C}^{(N+1)^2\times Q}$ is a frequency dependent spatial encoding matrix, calculated as

$$\mathbf{W}(f) = \mathbf{W}_{\text{EQ}}(f)\mathbf{Y}^{\dagger}, \tag{2.8}$$

where $\mathbf{W}_{\text{EQ}} \in \mathbb{R}^{(N+1)^2\times(N+1)^2}$ is an equalization matrix that compensates some of the effects produced by the sensors arrangement, † denotes the Moore-Penrose pseudo-inverse operation, and $\mathbf{Y} \in \mathbb{R}^{Q\times(N+1)^2}$ is the spherical harmonic encoding matrix, which contains the spherical harmonic weights for the direction of each sensor [24], [25].

[1] Time-frequency domain signals can be obtained using methods such as Short Time Fourier Transform (STFT) or filterbanks.

The spherical harmonics in $\mathbf{Y}$ are frequency independent and are calculated as

$$\mathbf{Y}(\Omega_Q) = \begin{bmatrix} Y_0^0(\Omega_1) & Y_{-1}^1(\Omega_1) & \dots & Y_n^m(\Omega_1) \\ Y_0^0(\Omega_2) & Y_{-1}^1(\Omega_2) & \dots & Y_n^m(\Omega_2) \\ Y_0^0(\Omega_3) & Y_{-1}^1(\Omega_3) & \dots & Y_n^m(\Omega_3) \\ \vdots & \vdots & \vdots & \vdots \\ Y_0^0(\Omega_Q) & Y_{-1}^1(\Omega_Q) & \dots & Y_n^m(\Omega_Q) \end{bmatrix},$$

where $\Omega_q = (\theta, \phi, r)$ are the locations of the sensors as mentioned before.

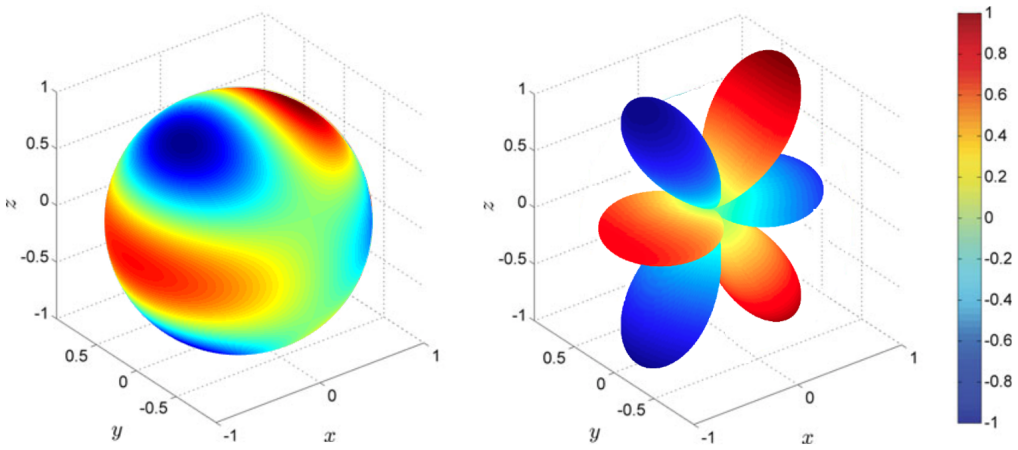


Figure 2.3: (left) Plot of spherical harmonic function of 3rd-order $Y_3^1(\theta, \phi) = -\sqrt{\frac{21}{64\pi}}(5\cos^2\theta - 1)\sin\theta e^{i\phi}$ over the surface of the unit sphere. (right) balloon plot for the real part of the same function with the distance to the origin defined by $|f(\theta, \phi)|$, and with red shades representing positive values and blue shades representing negative values of f. Adapted from [27] and [24].

2.2.2 Beamforming

After transforming the microphone signals into the spherical harmonic signals it is possible to define spatial filters in order to extract information from specific directions. This process is called beamforming, and it is a frequency independent procedure than corresponds to a weighted sum of the transformed spherical harmonic signals. Beamforming can be performed as

$$y(t, f) = \mathbf{w}^{\mathbf{H}}\mathbf{s}(t, f), \tag{2.9}$$

where $\mathbf{w} \in \mathbb{C}^{(N+1)^2 \times 1}$ is the steering vector, which is defined as

$$\mathbf{w} = \mathbf{y}(\Omega) \circ \mathbf{d}, \tag{2.10}$$

where $\mathbf{y}(\Omega)$ are the spherical harmonic weights for the direction of the beam, $\circ$ is the Hadamard product, and $\mathbf{d}$ is a vector of weights that defines certain properties of the beam [28]. In the simplest case, $\mathbf{d} = [1, 1, ..., 1] \in \mathbb{R}^{(N+1)^2 \times 1}$.

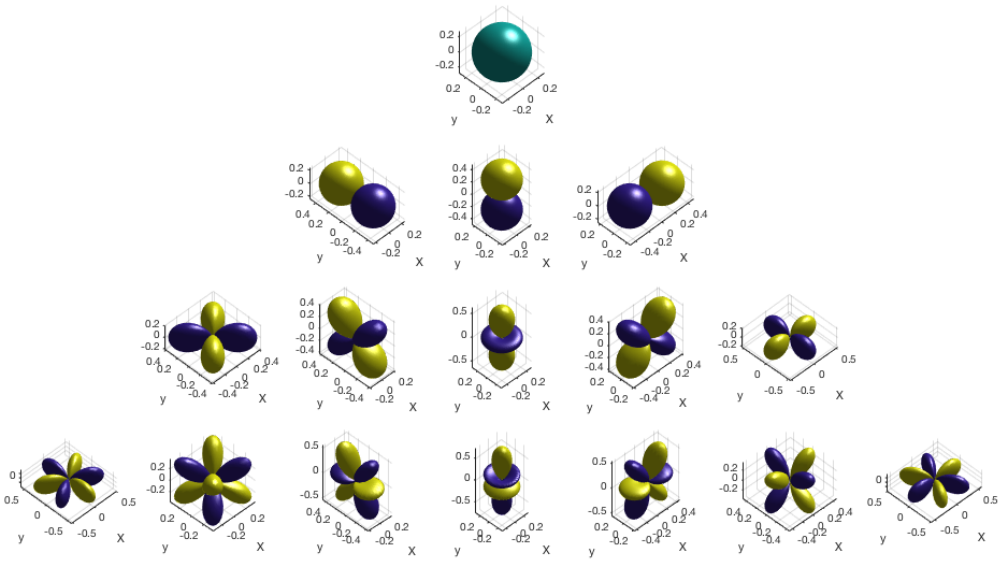


Figure 2.4: Balloon plots for the spherical harmonics basis functions for order $n = 0$ (top row) to $n = 3$ (bottom row). The colors indicate the sign (phase) of function, with yellow/green is positive, while blue is negative.

Using this method, beamforming is performed by defining a rotation matrix of the spherical harmonics. The maximum directivity of the beam will be determined by the order of the spherical harmonics. Figure 2.5 shows hypercardiod beams for orders one to three. In this case, higher orders offer better directionality with a narrower main lobe, but increased pattern complexity for the side lobes [29].

Finally, multiple beams can be used at the same time to cover a larger portion of the space. For example, in figure 2.6 two 3rd-order hypercardiod beams have been superimposed in the same plot.

In practice, there is a regularization parameter embedded in the equalization matrix, that influences the frequency bandwidth of usable components and spatial selectivity [26], [30], at the cost of amplified sensor noise. For auralization applications, lower noise is usually preferred as high audio quality is desired. Whereas for other cases, higher spatial resolution may be more preferrable.

2.3 Machine Learning

The main idea behind machine learning is to let machines learn how to solve a task, without having to explicitly explain the method behind the solution. This concept is similar to how humans learn, where for example, infants develop language skills not by reading grammatical rules but by imitation and practice. Thus, instead of defining rules, data is fed into a computer, so that patterns can be extracted, and the rules that define task are learned implicitly.

More formally, machine learning are algorithms that utilize experience E, to

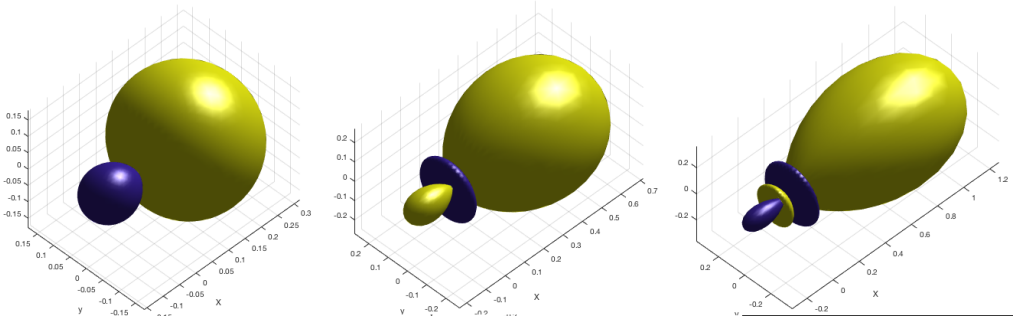


Figure 2.5: Balloon plots for hypercardiod beams of order $n = 1$ (left), $n = 2$ (middle), and $n = 3$ (right). The colors indicate the sign (phase) of the function, with yellow/green is positive, while blue is negative.

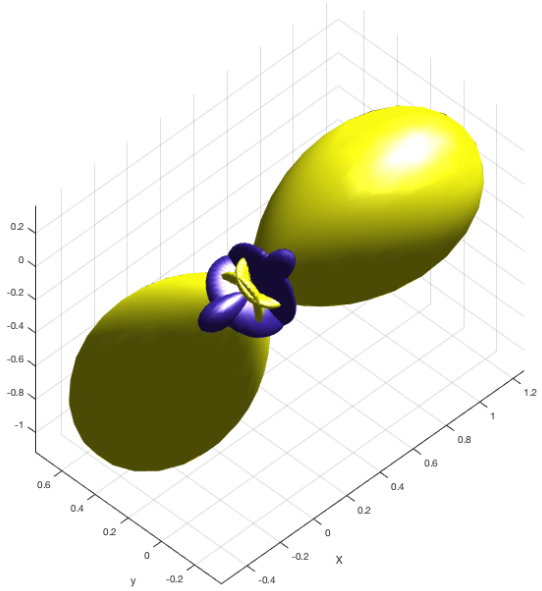


Figure 2.6: Balloon plot for two super imposed 3rd order hypercardiod beams pointing at $(\theta_1 = 0°, \phi_1 = 0°)$ and $(\theta_2 = 90°, \phi_2 = -45°)$. The colors indicate the sign (phase) of function, with yellow/green is positive, while blue is negative.

improve performance measure P, for task T [31]. The task should be clearly defined: translating a text between two languages, finding faces in a photograph, or transcribing audio into sheet music, are all examples of tasks. Furthermore, each task should include a performance measure, such as classification accuracy, prediction error, or distance to an objective. Lastly, the experience will be conformed of data, where multiple observations that are relevant to the tasks are collected to form a dataset.

This definition of machine learning is remarkably different from a general artificial intelligence, as the goal is not to create a sentient, autonomous program than can think and understand the world as a human being; rather, the goal is simply to maximize an objective parameter, for a particular application. Therefore, machine learning can be explained as applied statistics to approximate functions and solve optimization problems. This distinction becomes more blurry as tasks get more

involved.

The basic workflow of a machine learning solution is to first use raw data to extract meaningful features; relevant numerical or categorical values that describe the problem [2]. Then process these features to extract patterns, finding relationships between inputs and outputs. Complex systems can build complex features from simple ones, and then find more intricate patterns.

Although there are many different types of tasks that can be solved with machine learning, most can be generally explained as either **supervised learning**, where each data sample has features (input variables) and labels (output variables), or **unsupervised learning** where there are only features, and no labels.

2.3.1 Supervised Learning

In supervised learning, each data sample has features and targets (also know as labels). Features are independent variables that characterize each sample, whereas the targets define some property that can be interpreted as dependent variables that are the output of a function, when the features are used as input. Thus, on most tasks of this type, the goal is to let the machine learning algorithm learn the underlying function. To do this, a dataset containing both features and targets is used to train a model, so that it can then predict targets for inputs that it has not seen before [32].

The input features can be real valued numbers, integers, categorical or nominal values. Additionally, these features can be arranged in various ways: as a single vector concatenating all of them, as 2D matrix (such as images), or as more complex tensors, such as multichannel 2D matrices. On the other hand, output targets are usually numerical or categorical, and there can be either a single target or multiple. Problems with numerical targets are **regression** tasks, while categorical targets are **classification** problems.

As a general rule, problems with a large number of features (high dimensionality) require more complex models to solve them. Also, the main drawback of supervised learning is that annotating the data, which means defining the labels, is usually costly as it requires human input [33].

Regression

A regression task is one where the goal is to predict one or many numerical, continuous values, given some input. More formally, the goal is to build a model than can approximate a function f, such that with some inputs x it is possible to obtain outputs $\hat{y}$, so $\hat{y} = f(x)$. Ideally, the prediction equals the true target, $\hat{y} = y$. Additionally, it is possible to predict multiple values at once, so that $f : R^d \rightarrow R^m$, where d is the dimensionality of the input data (the number of features for each observation), and m is the dimensionality of the outputs.

[2] It should be noted that the feature extraction process can be part of the automatic learning, which means that the ML algorithm learns how to extract relevant features at the same it learns how to process them. This is know as end-to-end learning

As an example, it is possible to predict the house prices based on features such as size, location (coordinates), number of rooms, etc. In this case, the prediction is a single value (price) and the input features are numbers or categories that describe houses [33]. For this thesis, we define the problem of predicting room acoustics parameters as a regression task. So that the input x is the input signal, e.g. an impulse, while $f(\cdot)$ is the room and everything inside it, which transforms the input x into the output y, which are the acoustical parameters.

2.3.2 Unsupervised Learning

For some tasks, the data samples have features but there are no clear targets, either because the it makes little sense for the tasks, or because finding the correct targets is costly. Under these circumstances, it is possible to analyze the data in an unsupervised way, where the goal is not to predict some value but to extract information about the overall structure of the data. This information can then be studied further, such as finding which data samples are similar to each other. In general, in supervised learning the model learns the probability distribution of the targets, conditioned on the inputs, while in unsupervised learning the model learns the underlying probability distribution that generated the original data [31]. In other words, the former deals with $P(y|x)$, while the later with $P(x)$, where y are the targets and x are the input features.

There are different types of unsupervised learning tasks. For example, by detecting similarities between data samples, it is possible to assign clusters or groups to the samples. This is akin to a classification task without previous information about the number or meaning of the classes. Additionally, learning the full data generating distribution enables models that can synthesize new, artificial data samples that resemble the original ones.

Unsupervised learning algorithms can also be used to reduce the dimensionality of the data, by finding a simple, compact representation. The main idea is to learn a mapping that maps the data samples to a lower dimension latent space, preserving some properties, such as the relative distance of the data points. This mapping is useful in two ways: 1) the reduced dimensionality of the data can improve the performance of other ML algorithms, as it removes features that are not very relevant to the task, and 2) collapsing the data to 2 or 3 dimensions allows for easy visualization of complex data.

There are many different methods to perform dimensionality reduction. Here we utilize two different methods, principal component analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE).

Principal Component Analysis

A basic unsupervised learning method is principal component analysis (PCA). Due to its simplicity, it is a good starting point to analyze a dataset and to do dimensionality reduction. PCA is a linear technique that learns a mapping from the feature space X to the latent space Z, which maximizes the variance of the transformed data.

In practice, PCA is done by the eigen decomposition of the covariance matrix of the input features after centering the data (so that it has zero mean). The eigenvectors are the principal components, which are used to construct a projection matrix W. Each principal component is a vector in $\mathbb{R}^d$, where d is the dimensionality of the original data. The projection from the the original space X to the latent space Z is done with the dot product of the data matrix x and W. Dimensionality reduction happens when a subset of the eigenvectors is used to form W. Additionally, the eigenvalues show how much variance of the data is explained by each eigenvector. Therefore, a matrix W built using the first k principal components, will preserve the variance determined by the cumulative sum of the first k eigenvalues, divided by their total sum [34].

The principal components themselves are the new features into which the data is projected to. These do not represent any meaningful quality of the data, unlike the original features which correspond to some property of the data. Thus, it is not possible to interpret the projection directly. On the other hand, the projected data is decorrelated which can improve the performance of other ML algorithms.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

A new, state of the art method to do dimensionality reduction is the t-Distributed Stochastic Neighbor Embedding (t-SNE). This method learns a parametric mapping between the original, high dimensional space X to the latent space Z while preserving the local structure of the data in the original space as much as possible [35]. Unlike PCA, this mapping is non-linear, which means it can preserve non-linear relationships of data more easily than other methods.

The details of t-SNE are beyond the scope of this thesis, but it is enough to say that it preserves structure by maximizing the distance between dissimilar data points, and minimizing the distance between similar data points [36].

2.3.3 Neural Networks

A **neural network** (NN) is a computational model that approximates some function f^*. For example, a regression problem can be defined as $y = f^*(\mathbf{x})$, where some output value y is produced by function f^*, given the input $\mathbf{x}$. In short, the neural network models the true function f^* with the mapping $y = f(\mathbf{x}; \theta)$, where θ are parameters that need to be learned from the data [31].

The basic unit that builds a neural network is the **neuron**. In each neuron, the input data $\mathbf{x}$ is weighted by some parameter θ [3]. Additionally, the output of the neuron is fed into an **activation function** which is a mathematical function that introduces some properties to the network. The full operation of the neuron is shown in figure 2.7, and is defined as

$$\mathbf{y} = f(\mathbf{w}^T\mathbf{x} + b), \tag{2.11}$$

[3]For the neural networks used on this thesis, the set of parameters θ is composed of weights $\mathbf{w}$ and bias b.

where $\mathbf{x}$ is the input data, $\mathbf{w}$ are the weights (parameters), b is bias, both learned by the network, $\mathbf{y}$ is the output, and $f(\cdot)$ is the activation function. In this case, there are as many $\mathbf{w}$ parameters as features for the input $\mathbf{x}$, while the bias term b is a single value. Thus, the basic element of a neural network is nothing else but a linear combination of the input values, which are then shaped by some activation function. By using non-linear activation functions, the network can learn non-linear relationships between inputs and outputs. Examples of activation functions are shown in figure 2.8.

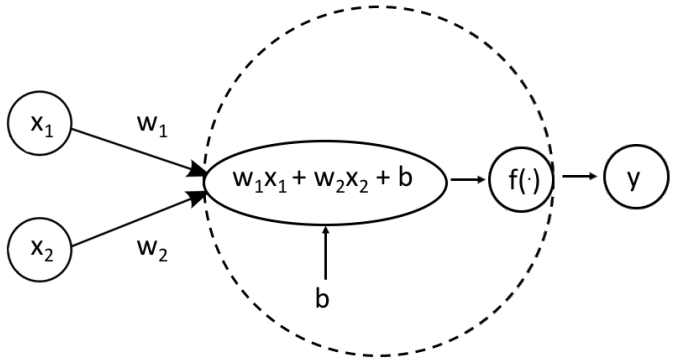


Figure 2.7: Schematic for the operations of a single neuron in a neural network. The dashed line marks the boundary for the neuron, which computes a linear combination of the input $\mathbf{x}$, weights $\mathbf{w}$ and bias b, that are then fed into an activation function f.

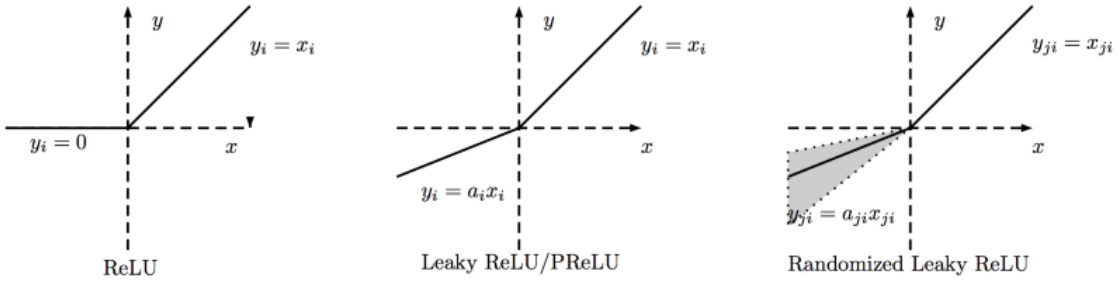


Figure 2.8: Example of common activation functions used to introduce non linearity to a neural network.

The true power of a neural network arises from the combination of multiple of these basic neurons. First, neurons can be stacked side by side, where the same input data is fed into multiple neurons, each with its owns particular set of weights $\mathbf{w}$ and bias b. This stack of neurons is called a **hidden layer**, and every neuron is referred to as hidden unit, or simply unit. Furthermore, the network can grow by adding multiple hidden layers sequentially, where the input of each hidden layer, is the output of the previous hidden layer. The amount of units per layer define the width of the network, whereas the total number of hidden layers define its depth.

Finally, it is also possible to connect units in any way, for example, skipping layers. The choice of width, depth, connections, and activation function (per unit) define the **architecture** of the network.

There exist a large variety of neural networks architectures. A **fully connected** (FC) neural network, also know as **multi layer perceptron** (MLP), or **feedforward neural network** is the fundamental neural network model. In the FC network, all units from one layer are connected to all the units of the next layer, and nothing else. This means that there are no recurrent connections, or connections that skip layers. It has been proven [33] that a FC network with non-linear activation functions, can be used as a universal function approximator, given enough width and depth. A diagram for a FC network is shown in figure 2.9.

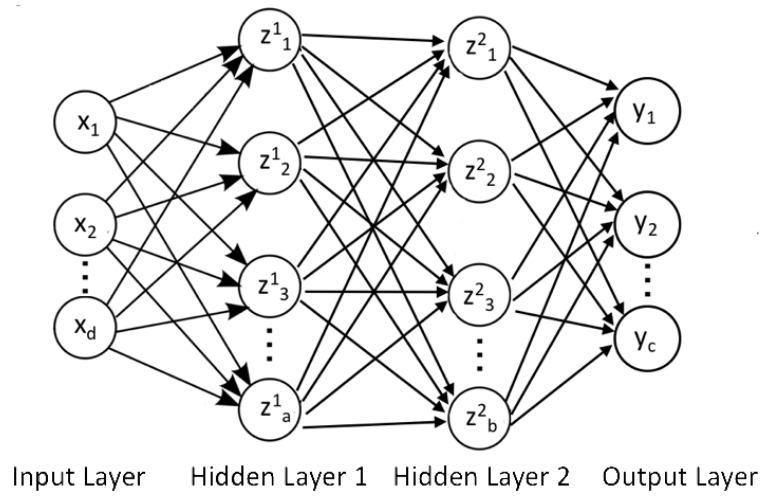


Figure 2.9: General architecture for a Multi Layer Perceptron (MLP), also known as fully connected (FC) neural network. The input layer $\mathbf{x}$ has d features, the hidden layers $\mathbf{z^1}$ and $\mathbf{z^2}$ have a and b units respectively, and the output layer $\mathbf{y}$ has c different targets.

It is important to note that there are two types of unknown variables at this point. On one hand, the values for all weights $\mathbf{w}$ and biases b of all layers are the parameters that the network will learn after training. On the other one, the architecture of the network is a choice that needs to be selected beforehand. This architecture is defined by hyperparameters, which are not learned by the training itself but need to be obtained by a different process. In short, a **learning method** is used to learn the parameters, while an **hyperparameter optimization** procedure is used to select the best architecture. Both of these variables play an important role on determining the effectiveness of the network, that is, how well can it approximate the true function f^*.

Learning Methods

A neural network approach to approximate a function $f(\cdot)$ is not much different than any other machine learning method. The main elements of this solution are: model,

cost function, and optimization algorithm. In this case, the model is the NN, and the optimization algorithm will be used to minimize some cost function in order to learn the parameters θ that define the network. In practice, the parameters are learned via a gradient descent algorithm.

The **cost function** is a mathematical expression that evaluates the performance of the model. For a regression problem, this cost function can be the difference between the predicted and true values. This difference is also known as **loss** or **error**. More formally, the cost function is

$$J(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim p_{data}} L(f(\mathbf{x};\theta), y), \tag{2.12}$$

where L is the loss function for a single data observation, $f(\mathbf{x};\theta)$ is the prediction generated by the model given input $\mathbf{x}$, using current set of parameters θ, and y is the true value or target. Thus, the total cost function is the expectation of the individual loss for all data.

The learning procedure is then the process of finding the values for parameters θ that minimizes $J(\theta)$. This is typically done using some variant of **iterative gradient descent** algorithm. The main idea is to evaluate the cost function using the current parameters θ and available data. Then compute the gradient of the the cost function, and move θ a small amount towards the direction of the negative gradient. The update rule for a basic gradient descent algorithm can be expressed as

$$\theta = \theta - \epsilon \nabla_\theta J(\theta), \tag{2.13}$$

where ϵ is the learning rate and $\nabla_\theta J(\theta)$ is the gradient of the cost function.

There are other, more sophisticated solvers than can be used instead of the basic gradient descent algorithm. In this work, all networks are trained using Adam. Adam adds an adaptive momentum term that modifies how the update for parameters θ are done. In short, the update is larger when the gradient over a set of iterations points to the same direction, whereas if the gradient is noisy, the updates become smaller [37], [31]. The details of this solver are out of scope of this thesis.

Hyperparameter Optimization

The hyperparameters of a model are selected by some optimization procedure. The main idea is to split the available into 2 subsets, called training and validation. Then, a model is build using a set of hyperparamters, trained using the training data, and evaluated using the validation set. This process is repeated using different sets of hyperparameters. At the end, the hyperparameter set that performs the best on the validation set is chosen. The validation set estimates the performance of all models in unseen data.

In practice, this search for optimal hyperparameters can be a simple method or a much more complicated optimization problem on its own. Simple approaches do a grid search over a space of possible values for all hyperparameters combinations. Other methods perform a random search, using random sets of hyperparamters.

Capacity, Overfitting and Regularization

In general, the **capacity** of a neural network is defined by the total amount of learnable parameters θ that are in the network. Usually, networks with higher capacity are able to approximate more complex functions, given enough training time. This means that models with higher capacity will be able to explain the training data better, and will therefore have smaller error than simpler models. In fact, it is possible to have zero error on training data, with a powerful enough model [31], [33].

However, the performance of the model in the training data is not the most important evaluation criteria. After all, the true values of the training data are known. What we care about is the performance of the model on data it has not seen before. Ideally, the network will approximate the true function $f^*(\cdot)$ for all data, and not just the data used for training. Furthermore, the network should not learn particularities of the training data that are not present on all the data, such as noise, or problems due to outliers. In other words, we want to minimize the **generalization error**.

In practice, this means that the ideal capacity of a model should be high enough to model the training data, but not too high, so that it fails the model the test data. An example of this concept is shown in figure 2.10, where a model with different capacities is fit to some training data.

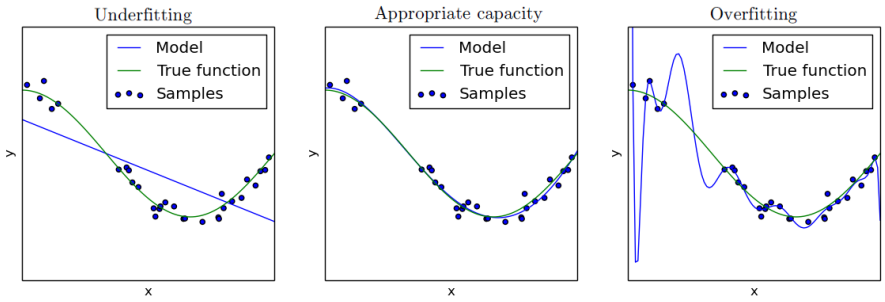


Figure 2.10: How the capacity of the model affects the fit to the data. (Left) Whne the model is too simple and has little capacity, it fails to approximate the function of the data, so the error between the model and the data is big. (Middle) A model with enough capacity can model most of the training data with small errors. This model is just complex enough to explain the data but not more. (Right) A model with too much capacity, will fit all the training data with no error, but will most likely fail to explain data not seen during training. Adapted from [38].

In order to minimize the generalization error, it is possible to limit the model capacity directly, for example, by carefully selecting the hyperparameters of the model. Another option is to add a term to the cost function used during training, that explicitly penalizes models that are "too complicated". This is a form of **regularization**. For example, by penalizing the norm of the weights vectors $\mathbf{w}$ learned by the network, the training will prefer solutions that have small weights. In the end, the best solution learned will balance the error between the predicted and

the true data, and this regularization term. This balance can be controlled by an hyperparameter.

There are many regularization methods, but the main idea across all of the them is to let the training procedure prefer some solutions over other, by adding additional constraints, either to the cost function, or the training procedure itself. If applied to the cost function, the general form is

$$\widetilde{J}(\theta; \mathbf{x}, \mathbf{y}) = J(\theta; \mathbf{x}, \mathbf{y}) + \lambda \Omega(\theta), \tag{2.14}$$

where $J(\cdot)$ is the cost function defined by parameters θ, evaluated for inputs an outputs $\mathbf{x}$ and $\mathbf{y}$, $\Omega(\cdot)$ is the new regularization term, and λ is the hyperparameters that controls how much penalty is applied. In this thesis, all models use the squared error with L^2 regularization, defined as:

$$\widetilde{J}(\mathbf{w}, b; \mathbf{x}, \mathbf{y}) = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{n}\sum_{i=1}^{n}\Big(\mathbf{y} - f(\mathbf{x})\Big)^2. \tag{2.15}$$

Chapter 3

Data Analysis

One of main goals of this work is to analyze the acoustical properties of a room and to build a computational model that is able to predict these properties based on geometrical and acoustical information without recurring to simulations. For this purpose, a dataset was composed. On this chapter, first, the data is described, including the collection process and some general qualities. Secondly, the data manipulation and preparation processes are explained. Finally, an exploratory data analysis is detailed, that exemplifies some of the dataset most peculiar characteristics.

3.1 Public Datasets

Machine learning algorithms are powered by data, and the performance of these algorithms on any given task is limited by the quality and quantity of this data. In the words of Andrew Ng, "if a ML algorithm is a rocket engine, data is the fuel that feeds the engine" [39]. Therefore, a successful ML solution requires not only sophisticated models and methods, but also adequate and abundant data. Moreover, most models can benefit from additional data, as long as they have enough capacity to exploit it. Even for tasks than can be accurately represented, such as physics based phenomena, large amounts of data is helpful [40].

In practice, developing large, diverse datasets has played a significant role in the results achieved in many fields. In computer vision, a major breakthrough occurred when the Imagenet dataset became popular [41], where state of the art solutions can now outperform human level in tasks such as object detection. In music classification, efforts like the Million Song Dataset (MSD) [42], or more recently the Free Music Archive (FMA) [43] have contributed to major improvements [1].

For room acoustics, there are no large, open, standard datasets that researchers from different institutions use, share and compare results. Instead, most studies collect their own data according to their needs. And although it is common to share this data with the scientific community, most of the time it has limited use beyond the scope of the original study, due to small size, lack of variety, or features types.

[1] Although music classification in general is far from solved, and more recently researchers are moving onto related tasks such as tagging.

For example, in [44] a set of room impulses is provided. It has 3 different rooms, including a medium hall with 200 seats, and big, open space that works as a library, and a small classroom. The data includes multiple loudspeaker positions for each room, 2 microphone positions, and both omninidirectional and B-format impulse responses. However, the dataset only includes basic geometric information about each room, such as general dimensions, and positions for loudspeakers and microphones. There is no data about acoustical properties of the rooms.

Other datasets focus on different problems, such as sound event detection inside noisy environments. For instance, [45] and [46] provide a collection of sounds and room impulses in ambisonics format. Much of the data variety comes from the event themselves and not the responses. Simulated room impulse responses are used in [47], although for very few examples.

For this thesis, we are interested in the prediction of general room acoustics properties based on different features such as the dimensions of the room or the furniture inside. None of the aforementioned datasets provide both, high quality multichannel room impulse responses for a variety of positions and acoustical environments, and detailed information about the geometry and absorption coefficients of environments and objects. Therefore we analyze a novel dataset, produced by the Department of Signal Processing and Acoustics of Aalto University [2]. The data consists of real-life acoustical measurements and full 3D geometrical modelling of the room and furniture, in addition to absorption coefficients for most elements. Even though the size and variety of the data is not cosidered large, especially when it is compared to what may be obtained with simulations, the data offers a true representation of a real world phenomenon, including all the noise and disturbance associated with real world measurements. For these reasons, the data is suitable for a proof of concept.

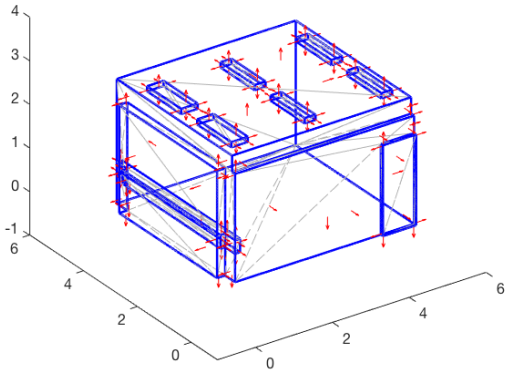
3.2 Room Acoustics Dataset

The dataset consists of a total of 3332 data records [3], that include audio measurements and other properties. All measurements were made in a single room at the Otaniemi campus of Aalto University during the summer of 2017. The room is a small office, with carpeted floor, light acoustic treatment in the ceiling (in the form of perforated gypsum panels), a large window on one wall, and a single door. The dimensions of the room are show in table and figure 3.1. A total of 4 different fixed loudspeaker positions and 1 microphone position were used. For each observation, the room impulse response (RIR) was obtained by reproducing an exponential sine sweep with a loudspeaker inside the room, and then recording the sound with a microphone [48]. The microphone used is an spherical microphone Eigenmic with 32 capsules, so for each measurement there are 32 recordings or channels.

Even though all the data was obtained from a single room with fixed loudspeaker and microphone positions, variance was introduced by manipulating the objects

[2] It should be noted that the data collection process was not part of this thesis itself, as it was carried out by other personnel of the department

[3] In this work the terms data record, observation, and example are used interchangeably.



Property	Value
Width	4.398 m
Height	2.856 m
Depth	4.867 m
Volume	61.13 m^3
Wall surface	95.73 m^2

Figure 3.1 & Table 3.1: Dimensions of the room used for all the acoustical measurements in the dataset.

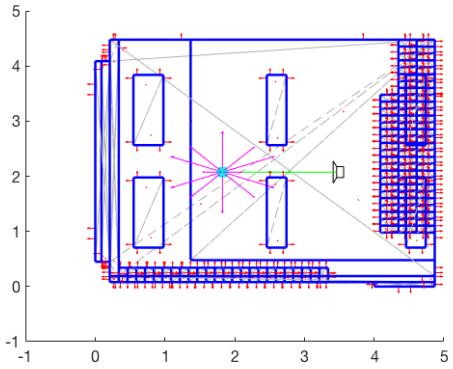
placed inside the room. The room itself is rectangular and mostly empty, with only a few surfaces that break the flatness of the walls: lamps in the ceiling, and an edge around the window. The objects added to the room included shelves, crates and most noticeably highly absorptive wedges, which are commonly used in anechoic chambers. In the end, the placement and position of these objects in the room creates a unique acoustical environment. In this work we refer to the combination of room, furniture and objects as room configuration. In total, there are 833 different room configurations available.

A data observation collected for each room configuration, and a single loudspeaker position out of the 4 available, thus there are 3332 observations. Each data record includes the full 3D model of the room configuration, a 2D equirectangular projection of a spherical (360° panoramic) photograph of the room configuration at the time of the measurement from the microphone location, position and orientation coordinates of the loudspeaker and microphone used, and finally the 32 channel RIR. A summary of these raw features is shown in table 3.2. The sampling frequency used is 48 000 Hz.

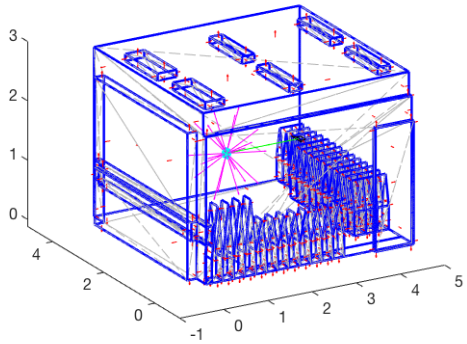
Feature	Type	Dimensionality
3D model	complex	N/A
Spherical photograph	image	2688 x 5376 x 3
Loudspeaker position and orientation	vector	1 x 3
Microphone position and orientation	vector	1 x 3
Room impulse response	multichannel sequence	240000 x 32

Table 3.2: Features available for each raw data observation.

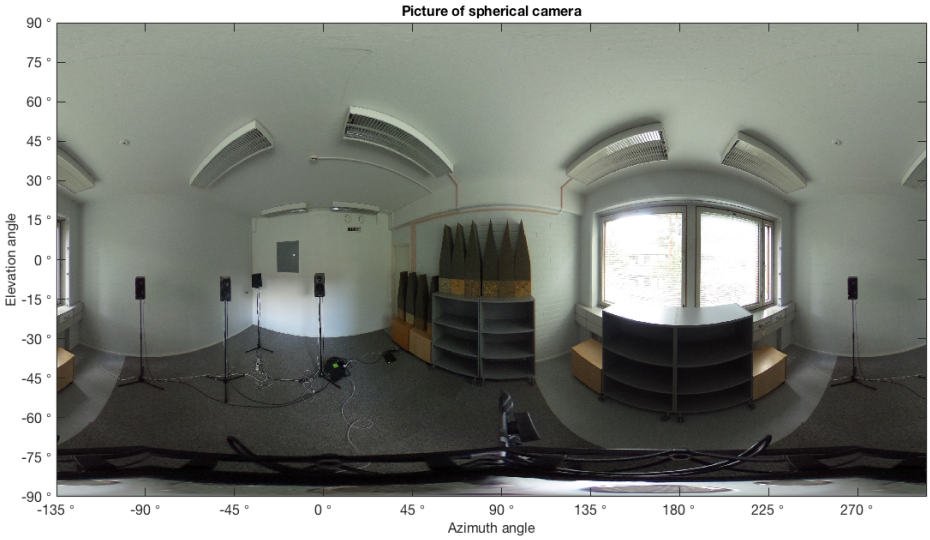
A visualization of the raw data for a typical room configuration is shown in figure 3.2. The 3D model is rendered from two different perspectives, and it includes the room, objects inside, and the loudspeaker and microphone locations. For this configuration, most of the room in empty, except for some absorptive wedges placed



(a) Top view



(b) 3D View

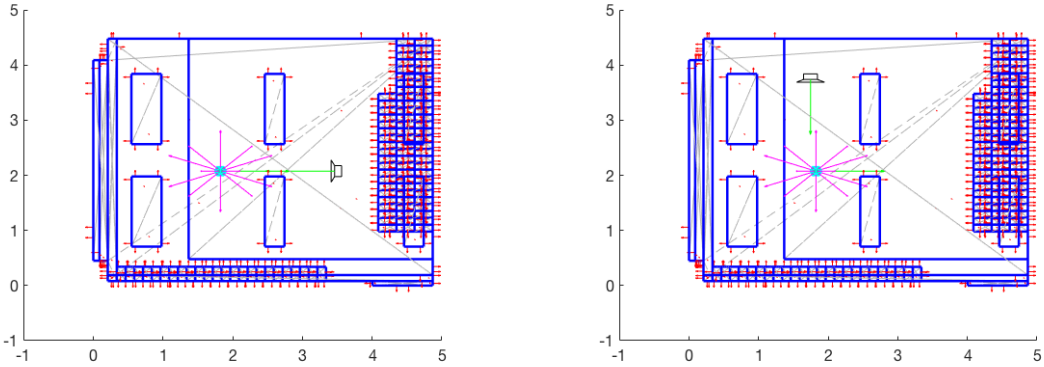


(c) 2D Spherical photograph

Figure 3.2: Geometrical data for a typical room configuration, including the general layout of the room along with the placement of fixtures such as lamps, furniture, or absorptive wedges. In the 3D models (a) and (b), the speaker icon represents the loudspeaker position, and the green line marks its orientation. The cyan sphere shows the microphone position, where each magenta line marks the orientation of the 20 faces of the icosahedron volume used to analyze the room. The spherical camera (c) shows the photograph of the same room configuration at the microphone position.

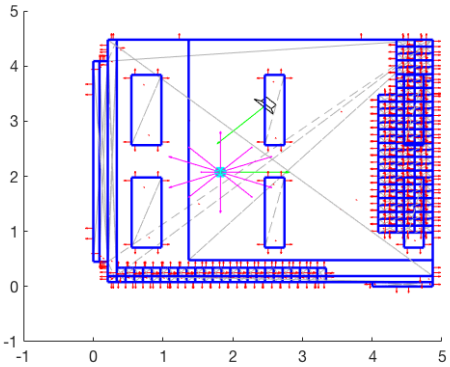
on the floor, along the front and right walls. The icosahedral volume is shown, where each magenta line marks the direction of each of its 20 faces. Only the position of the loudspeaker used in the data record is marked in the 3D model, however, the spherical photograph shows all the speakers at the same time.

Lastly, all the locations of the loudspeakers as shown in figure 3.3. This positions are fixed for all room configurations. For analysis, we refer to each of the positions by the labels *front* , *left90degrees* , *left45degrees* and *farAway*. Do note that all positions

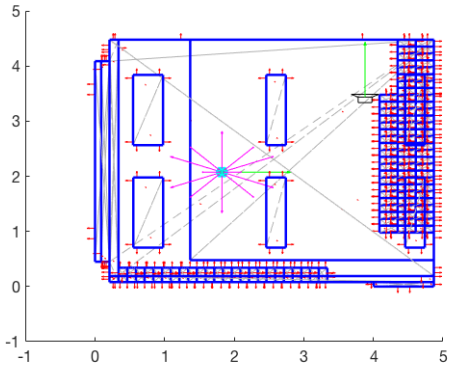


(a) *front*

(b) *left90degrees*



(c) *left45degrees*



(d) *farAway*

Figure 3.3: Location of the 4 different loudspeaker positions inside the room. These positions are fixed for all room configurations.

except for *farAway* point directly at the microphone.

3.2.1 Data pre-preprocessing

The raw data described before lacks two important attributes. First, there is no information about the absorption coefficients α of the room surfaces and objects, and second, room acoustic parameters need to be extracted from the RIR. So an additional procedure needs to be applied to the data to obtain the relevant features and targets. Here we call this process the **pre-preprocessing** of the data, in order to differentiate it from the preprocessing step that is done later, during the estimation phase, as a method to prepare the data for the ML models.

For spatial audio, we are interested in the direction where sound is coming from. In addition, the differences between sounds coming from multiple directions are also relevant. For room acoustics, the acoustical properties of the room configuration will affect sound from multiple directions differently. To capture the diversity among directions, we first consider an imaginary volume right at the microphone position and analyze both the input features and perceived sound for several directions originating from this point.

In practice, the pre-preprocessing procedure is as follows. For each data record, the 3D model of the full room configuration is loaded. Then an icosahedral volume [4] (see figure 3.4) is placed in the model, at the location of the microphone. Each face of the icosahedron points to a unique direction in the room. First the geometrical data is processed. For each of the 20 directions, the 3D model is analyzed, using ray casting, to identify the total absorption area (ignoring obstacles), visible absorption area (taking obstacles into account), distance to nearest surface and distance to farthest surface. Given than α depends on frequency, all absorption values are calculated for octave bands. This results in 4 new features sets, two of 20 x 6 values, and two of 20 x 1. We refer to each set as a **spherical map**.

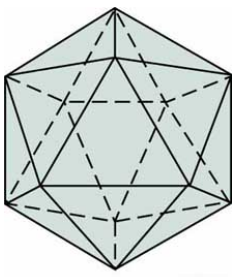


Figure 3.4: Icosahedron with 20 triangular faces. This polygon is used to extract directional information of the raw data.

Afterwards, the relevant acoustical parameters are computed for each direction of the icosahedron. To do this, the 32 channels of the RIR are loaded, and using

[4] An icosahedron is a polyhedron with 20 faces. A regular icosahedron has faces that are all equilateral triangles, and is one of the five Platonic solids [49].

microphone array processing we generate 20 different 3rd order hypercardiod signals, with them pointing to the same 20 directions as the center point of the faces of the icosahedron. The result is a spatially-constrained single channel RIR for each direction, which are used to calculate the values for T_{60} , EDT, C_{50} , DRR and L. The actual computation is done with the toolbox provided by University of Surrey [50]. All these parameters are computed in octave bands, except for C_{50} and DRR which are broadband. Table 3.3 shows a summary of all the available features after the pre-preprocessing step. At this point, some of the raw features are discarded from further analysis.

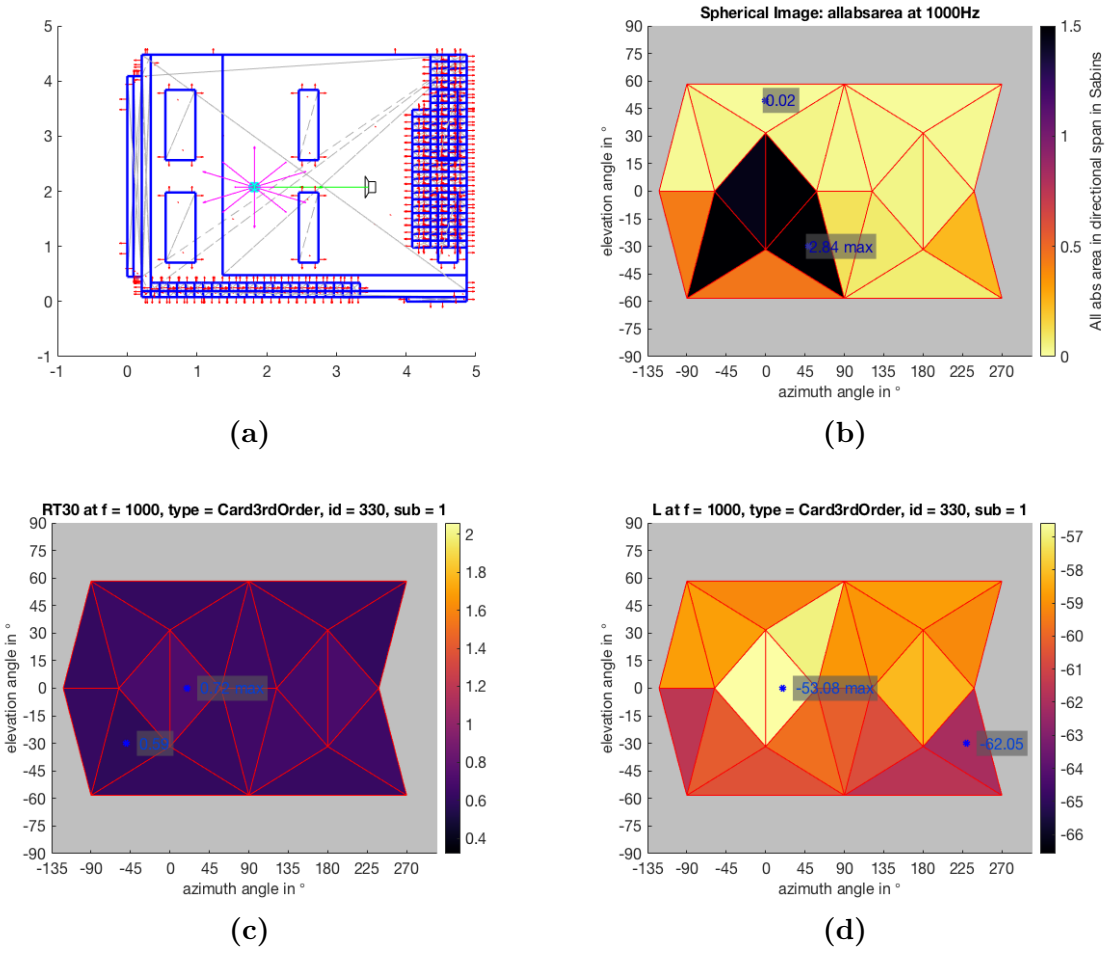
Feature	Type	Dimensionality	Units
All available absorption area	spherical map	1 x 20 x 6	sabine
Visible absorption area	spherical map	1 x 20 x 6	sabine
Nearest surface distance	spherical map	1 x 20 x 1	m
Farthest surface distance	spherical map	1 x 20 x 1	m
Spherical photograph	image	2688 x 5376 x 3	pixel
Loudspeaker position and orientation	vector	1 x 6	m
Microphone position and orientation	vector	1 x 6	m
T_{60}	spherical map	1 x 20 x 6	s
EDT	spherical map	1 x 20 x 6	s
C_{50}	spherical map	1 x 20 x 1	dB
DRR	spherical map	1 x 20 x 1	dB
L	spherical map	1 x 20 x 6	dB

Table 3.3: Summary of all features available for each observation after the pre-preprocessing step.

An example of data after pre-preprocessing for a typical room is shown in figure 3.5. The total absorption area for all processed directions computed at the microphone position are visualized as a spherical map (3.5 (b)), where each triangle corresponds to a face of the icosahedron. Thus, the highest absorption values for this configuration are found in the faces pointing front and down, where most of the wedges are located. The values for T_{60} and L of all 3rd order cardiod signals are also shown as spherical maps in 3.5 (c) and (d).

3.3 Exploratory data analysis

An exploratory data analysis (EDA) is a useful method to understand some of the properties of a dataset [51]. An EDA is usually the first step applied to a new dataset, where the goal is to find patterns, identify anomalies, or confirm assumptions. This process can discover areas that become future research opportunities. For example, it is common to visualize the distribution of features, or remove features that are highly correlated. In this section, a basic EDA evalutes some of the most salient attributes, without going into deep details.



(a) **(b)** **(c)** **(d)**

Figure 3.5: Visualization of some input features and acoustical parameter values for a typical room after pre-preprocessing the raw data. (a) top view of the 3D model of the room configuration, including the loudspeaker and microphone positions, and orientation. (b) spherical map of all absorption available in the room. (c) spherical map for T_{60} values. (d) spherical map for L values. All values shown for 1000 Hz.

Figure 3.6 shows the distribution of values for all available absorption area found in the room, for all observations, in the 20 directions analyzed, at a single frequency band. It is not surprising that some directions have a very small range of values, as these directions most likely point to areas of the room where no treatment is applied, such as those that point to the ceiling. Figure 3.7 shows a similar plot with the distribution of the computed values for T_{60} and L, for all hypercardiod signals. As expected, T_{60} has comparable distributions across all directions, as it is the result of late reflections coming from all directions. L is more varied, because given the fixed loudspeaker positions, some directions always have stronger direct sound than others.

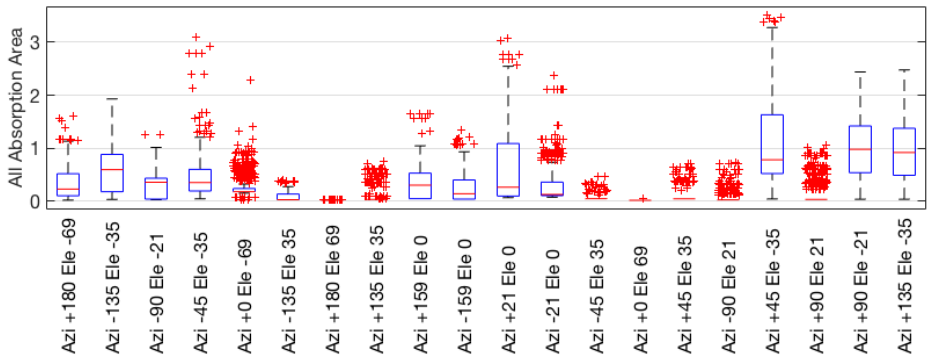


Figure 3.6: Distribution of values for all absorption area available in the room for every direction of the icosahedron, at 1000 Hz.

Figure 3.8 shows the relationship between visible absorption and T_{60} for all room configurations, a single frequency band, and all loudspeaker positions. There is little difference between the distribution of values across all loudspeaker positions. Moreover, there is no correlation between visible absorption and T_{60} value, as it can be seen that similar T_{60} values occur for a wide range of visible absorption areas. A similar analysis is shown in figure 3.9 for L values. Once again, there is no evident correlation between L and visible absorption. More interestingly, for all loudspeaker positions except *farAway*, there are 2 clear clusters of values for L. These highest values are most likely those positions that point directly towards the loudspeaker on each observation, as the direct sound usually has higher level than reflections. Therefore, as position *farAway* is not pointing towards the microphone, this difference is not observed.

An analysis of the distribution of T_{60} values and its relationship with all available absorption area, for all room configurations, loudspeaker positions and frequency band of 1000 Hz is shown in figure 3.10. On this plots, the mean values for all directions of the icosahedron were considered, as the goal is to analyze the distribution of values for each observation. As expected, when the mean of the total absorption present in the room is low, the mean T_{60} across all directions is high. In other words, when the room has little absorption, the reverberance is high. As mean total absorption

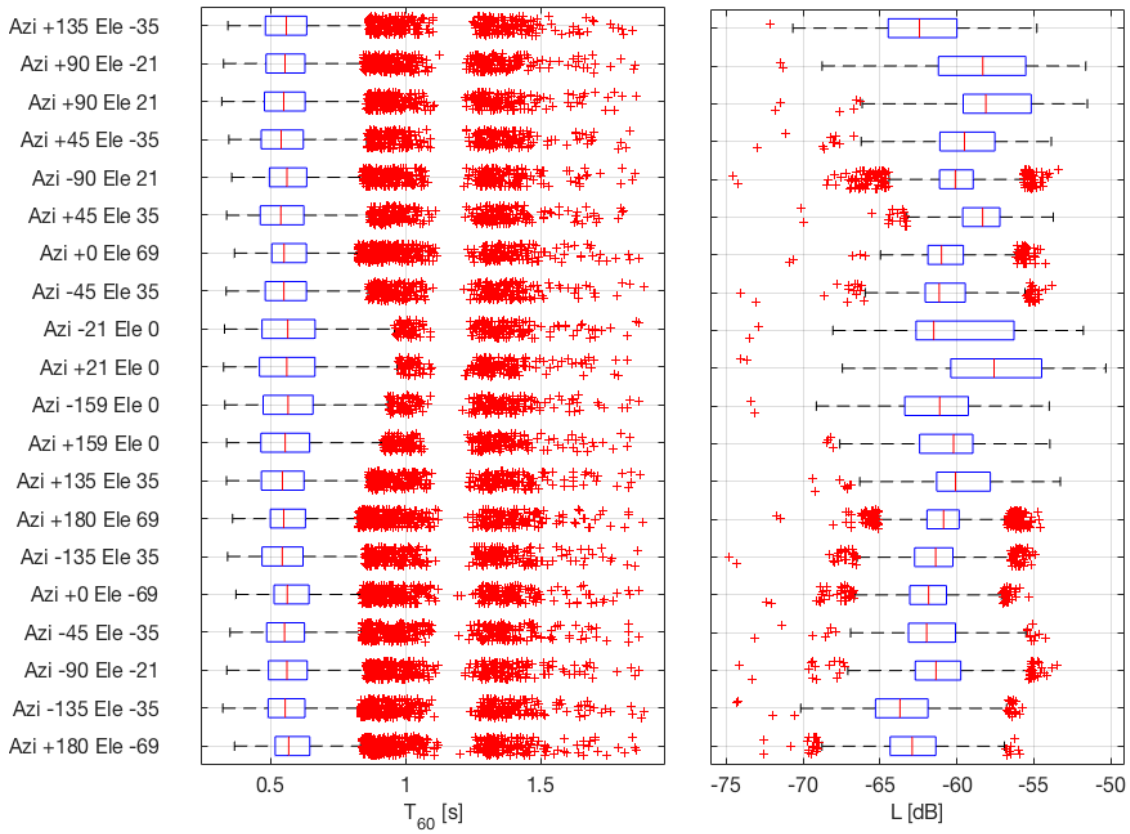


Figure 3.7: Distribution of values for all T_{60} (left) and L (right) for every direction of the icosahedrical volume, at 1000 Hz..

increases, the T_{60} decreases until it stabilizes. In addition, the variance for each room configuration is very small, which supports the idea that T_{60} has little difference across directions. There is minor deviation among loudspeaker positions, except for slightly more variance for position *front*, specially for certain room configurations.

The same analysis is shown in figure 3.11 for L values. The relationship between mean values of L and absorption is comparable with that of T_{60}, however, here the variance is much higher and follows an inverse trend than the mean, it increases with the mean absorption. L correlates to the sound pressure level of the direct sound, for this reason, the variance across direction is high.

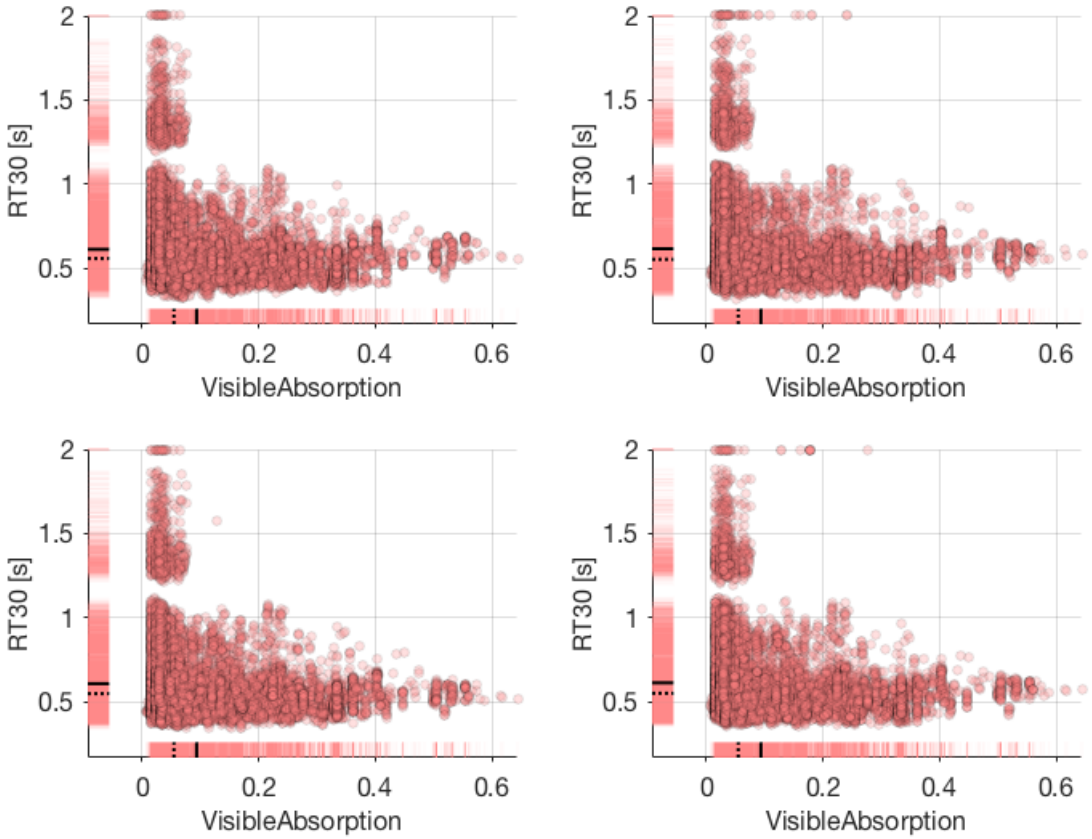


Figure 3.8: Relationship between visible absorption and measured T_{60} values for all room configurations at 1000 Hz, with loudspeaker positions *front* (top left), *left90degrees* (top right), *left45degrees* (bottom left), and *farAway* (bottom right). Each point in the scatter plot is a pair of visible absorption and T_{60} value for a single direction of the icosahedron volume faces placed in the microhpone position. Rugplots on top of the x and y axes show the marginal distribution, where the solid and dashed lines mark the mean and median respectively. Points where the $T_{60} > 2$ s where removed from the analysis as they are considered outliers.

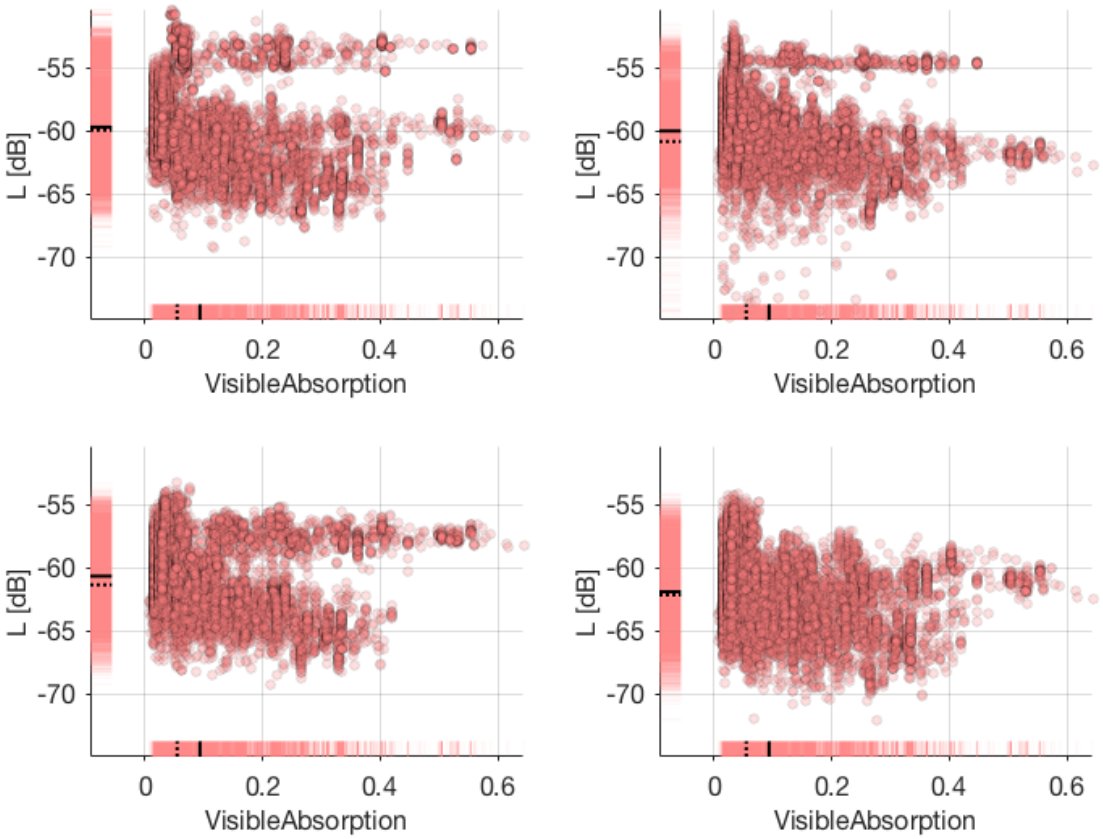


Figure 3.9: Relationship between visible absorption and measured L values for all room configurations at 1000 Hz, with loudspeaker positions *front* (top left), *left90degrees* (top right), *left45degrees* (bottom left), and *farAway* (bottom right). Each point in the scatter plot is a pair of visible absorption and L value for a single direction of the icosahedron volume faces placed in the microphone position. Rugplots on top of the x and y axes show the marginal distribution, where the solid and dashed lines mark the mean and median respectively.

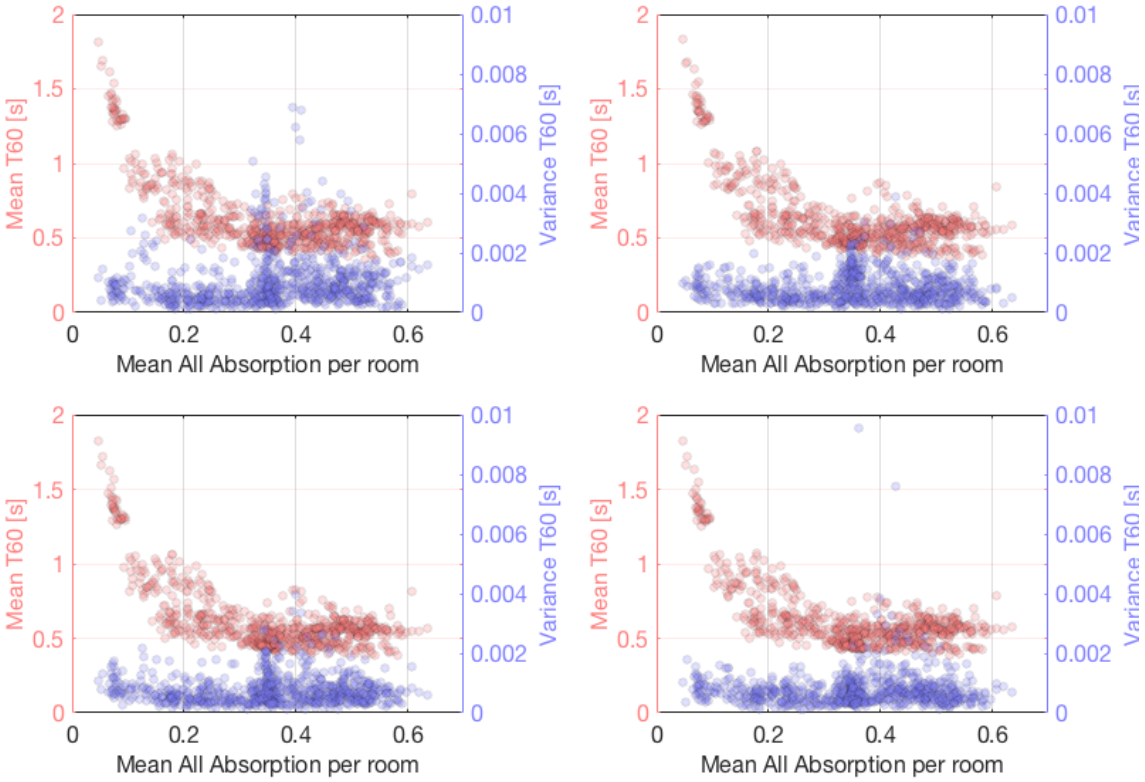


Figure 3.10: Relationship between the absorption present in a room configuration and the measured T_{60} values for all room configurations at 1000 Hz, with loudspeaker positions *front* (top left), *left90degrees* (top right), *left45degrees* (bottom left), and *farAway* (bottom right). The x axis is the mean of all absorption present in the room for all directions. Each point is the mean (red points) or variance (blue points) of the T_{60} values for all directions in a particular room configuration. The absorption values consider all the absorption present in the room and not just what is visible from the microphone position. Points where the $T_{60} > 2$ s where removed from the analysis as they are considered outliers.

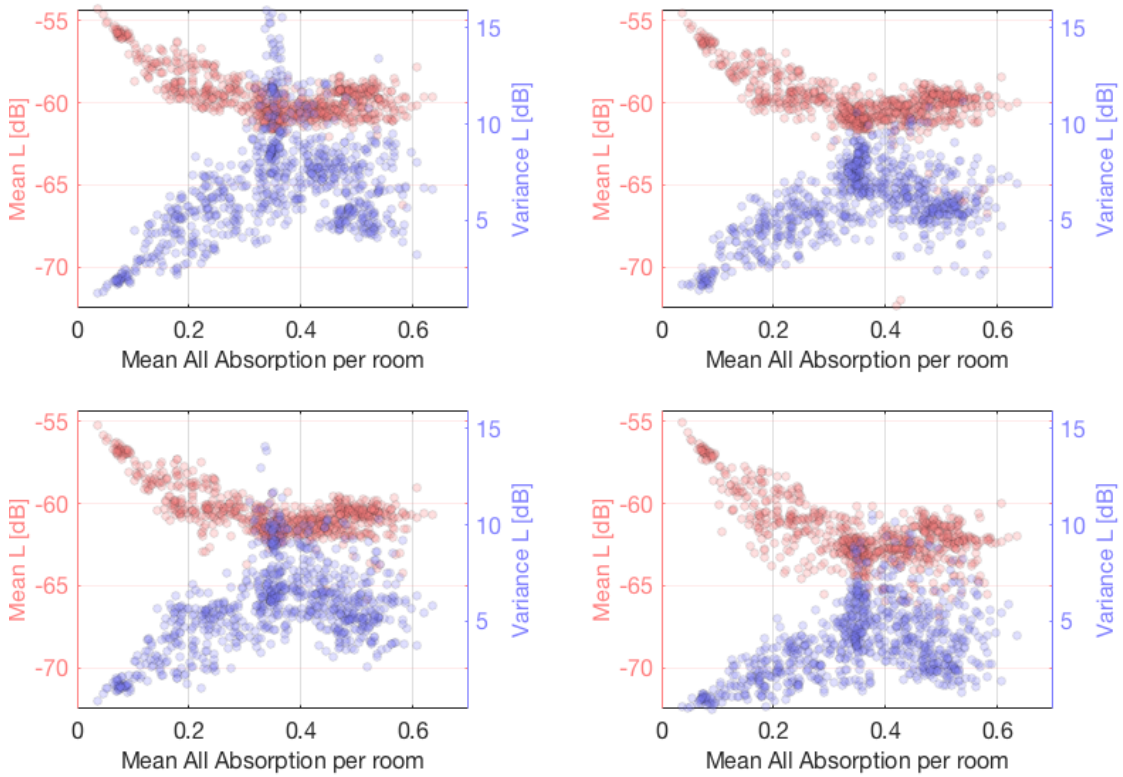


Figure 3.11: Relationship between the absorption present in a room configuration and the measured L values for all room configurations at 1000 Hz, with loudspeaker positions *front* (top left), *left90degrees* (top right), *left45degrees* (bottom left), and *farAway* (bottom right). The x axis is the mean of all absorption present in the room for all directions. Each point is the mean (red points) or variance (blue points) of the L values for all directions in a particular room configuration. The absorption values consider all the absorption present in the room and not just what is visible from the microphone position.

Chapter 4

Methodology

In this chapter, we describe the preprocessing procedure which the data (described in chapter 3) undergoes, where it is prepared for the ML models. Next, the performance metrics used to evaluate all models are explained. Finally, the models themselves are listed and detailed, including the hyperparameter optimization procedure employed to tune the models.

4.1 Data Preparation and Preprocessing

After the pre-preprocessing step described in 3.2.1, the data needs to be prepared to be used in the neural networks models. This means that a data matrix (or tensor) needs to be built selecting some or all the features available as inputs, as well as the appropriate targets as outputs. Additionally, the inputs need to be standardized.

Outliers Removal

One of the limitations of the method used to compute the T_{60} values is that sometimes it can give very wrong values. This is most likely because the background noise in the RIR was too high, which affects the regression of the energy decay curve. Thus, it is not uncommon for some T_{60} values to be as high as 20 s, which is severely unreasonable for the room used in the dataset.

Therefore, there is a need to remove from the analysis those observations where the T_{60} values are higher than some threshold t_{limit}. For most computations of this thesis, the threshold was set to t_{limit} =2 s. This threshold does not remove all outliers (as perhaps values larger than 1 s are already wrong), but going any lower would severely limit the amount of available data for models that predict multiple bands. In such cases, all observations with $T_{60} > t_{limit}$ for any frequency band were removed.

Then same rule was applied for models using EDT. No outliers where removed for the other acoustical parameters.

Normalization

For all models, the input data was normalized so that each feature has zero mean, and unit variance. That said, in most cases this normalization process was not very important, as the data has very similar ranges for most features. No further analysis was done to, for example, compare different normalization strategies.

Split

For all experiments, the data was split into different subsets after removing outliers unless noted otherwise. The typical split used 70 % of the data for training, 10% for validation, and 20% for test. Validation was only used as early stopping criteria when training the models. In addition, the data observations where shuffled before splitting, so that each run works on a different sample. Where applicable, cross validation was used using 5 folds.

Augmentation

Data augmentation was used to artificially increase the amount of training data and to combat overfitting. The method used was to create copies of the training observations, and add a small amount of white noise to the copies. For most experiment, the data was replicated 4 times, so that the training set has 5 times the total number of available observations. The added noise was sampled with Gaussian distribution with mean 0 and standard deviation of 0.2. A cursory analysis of the effect of augmentation showed that increasing the amount of augmentation did not improved the performance significantly, while higher standard deviations for the noise escalated errors. Only training data was augmented, as both validation and test data were left unperturbed.

4.2 Performance metrics

In any machine learning application it is important to define objective measures that determine the quality of solutions. These metrics allow to easily compare models and to correctly evaluate whether one is better than another [39]. However, in real world applications determining what is a better solution is not simple. Commonly, there are multiple aspects that need to be considered, and some of these values can be unrelated, or even inversely related.

For example, in a particular classification problem, finding the correct class for the input data (which means having high classification accuracy) is important. At the same time, having a system that makes the classification in a computationally efficient manner is also important, as the application might be deployed in consumer devices with limited resources. Therefore, the ideal solution would have high accuracy and low computational cost.

This thesis deals with the prediction (or estimation) of common room acoustics parameters as a regression problem. Thus, it is important to quantify the *regression*

error, to measure how much the predictions deviate from the true values. However, this can be done in multiple ways, and each method shows different facets of the errors. It is important to mention that these performance measures are not necessarily the same as the cost function used to train the ML models, as these measures can always be computed from the final predictions of any model.

Mean Square Error (MSE)

Perhaps the most common way to magnify the regression error is the **mean square error** (MSE) and the closely related **root mean square error** (RMSE), which are computed as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{4.1}$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{4.2}$$

where n is the total number of observations, and y_i, $\hat{y}_i$ are the true value and predicted value of the ith observation respectively.

Besides measuring error, the MSE can also work as an unbiased estimator of the population variance, by scaling it by $\frac{1}{n-1}$ [52]. However, as the MSE uses square values, it is sensible to outliers, where unusually large values are over penalized. Nonetheless, the MSE is a good starting point, and it is particularly insightful because it is commonly used as a cost function as it is easily derivable.

Mean Absolute Error (MAE)

The **mean absolute error** (MAE) is similar to the MSE, with the difference that instead of squaring the error, here the absolute value is used. It is compute as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{4.3}$$

where n is the total number of observations, and y_i, $\hat{y}_i$ are the true value and predicted value of the ith observation respectively.

The MAE is not as sensible to outliers as MSE, but it cannot be used to estimate the variance as well. Additionally, both MAE and MSE (or RMSE) depend on the distribution of errors. MAE is better suited when the errors are more or less uniform, as it assigns the same weight to all errors. On the other hand, if the errors are distributed normally, MSE might work best. Another disadvantage of the MAE (and MSE) is that it is not relative to the magnitude of the true values y, which makes it difficult to interpret.

Mean Absolute Percentage Error (MAPE)

The **mean absolute percentage error** takes into account the actual value of each observation, giving a scaled measure of error. Its main advantage is that it is very easy to interpret, as it shows how big the errors are on average as a percentage of the true values, while not being extra sensible to outliers. MAPE is computed following [53] as

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i + \epsilon} \right|, \tag{4.4}$$

where n is the total number of observations, ϵ is a small number to avoid division by 0, and y_i, $\hat{y}_i$ are the true value and predicted value of the ith observation respectively.

Regression Accuracy

All previous performances metrics measure the size of the error, where smaller values are better. The **regression accuracy** measures instead how close the prediction is to the true values, so that larger values are better. It is computed as

$$\text{accuracy} = (1 - \text{MAPE}), \tag{4.5}$$

The main problem with this metric is that it can be lower than 0, as the MAPE is not bounded for very large errors. On the other hand, when errors are small, the accuracy is a very clear way to compare results, and it should be as close to 100% as possible.

Finally, for models that estimate multiple values at the same time (for example, those that estimate all frequency bands in one pass), y and $\hat{y}$ are row vectors for each observation, and the total performance metric is the mean of all columns.

4.3 Machine Learning Models

4.3.1 Baseline

The baseline model is a FC network with 2 layers and 30 and 15 hidden units for each layer respectively. The activation function for both layers is relu, and there is no dropout. The training (for this and all models) was done using Adam optimizer [37] and the cost function is the MSE as defined in section 4.2 with L_2 regularization. The minibatch size was set to 256, using learning rate of 5 x 10^{-3} and L_2 coefficient λ of 1 x 10^{-4}. Weights are initialized using random values from Normal distribution with mean of 0 and standard deviation of 0.01, while initial biases are set to 0. The gradient threshold is set to 1, which means that all gradient values larger than 1 will

be clipped to 1 [1]. Additionally, the validation patience is set to 25 epochs, so that whenever the validation error stops decrease for 25 continuous epochs, training is stopped (early stopping). The hyperparameters for this model are summarized in table 4.1.

Group	Hyperparameter	Value
Model	# layers	2
Model	# hidden units (per layer)	[30, 15]
Model	activation (per layer)	{ relu, relu}
Optimizer	learning rate	10^{-3}
Optimizer	gradient threshold	1
Regularization	L_2	L_2
Regularization	dropout	$[0, 0]$

Table 4.1: Hyperparameters for the baseline model.

This baseline model is used to explore the prediction of different acoustical parameters using multiple input features. Therefore, even though the core model is the same, the input and output layers change depending on the choice of features and targets. We refer to this choice as input and output modes.

Input Layer

In chapter 3 the features available after pre-preprocessing the data were mentioned. In practice, for most of the experiments mentioned in this thesis, the input is the spherical map of all available absorption for a single frequency band. This means that the input layer has width of 20. Additional experiment were conducted using additional features. For instance, all available absorption and visible absorption ($allAbs$ + $visAbs$) with an input width of 20 + 20 = 40; or all available absorption, visible absorption, distance to nearest surfaces and distance to farthest surfaces ($allAbs$ + $visAbs$ + $nearSurf$ + $farSurf$) with an input width of 20 + 20 + 20 + 20 = 80.

Other modes include multiband features, where we stack the spherical maps consider of all frequency bands at the same time. For example, using only all available absorption and all frequency bands, the input width is $20 \times 6 = 120$. The largest mode uses all maps for all frequencies, so the total input width is $20 \times 6 + 20 \times 6 + 20 + 20 = 280$, as the distance to surfaces does not depend on frequency.

For the baseline model, each loudspeaker position was treated as a different dataset, primarily because there are only 4 different options. Therefore, there is no need to include features relating to these positions. This was done to facilitate

[1] Gradient clipping is usually only needed for very large neural networks, such as models with many layers or recurrent architectures. We applied gradient clipping as it helped to obtain consistent results

analysis and study, for example, if some speaker positions were easier to predict than others. The main drawback of this is that the amount of available data is reduced to only 833 observations for each loudspeaker (as there are 833 different room configurations).

Given the high dimensionality of the spherical photographs ($2688 \times 5376 \times 3 > 43$ million) compared to the size of the data, these images were not used in any models.

Output Layer

For the output layer, there are also multiple modes possible, and they depend on the input layer. When the input layer has values for a single frequency band, the output layer has width of 20, as it predicts the values for a single acoustical parameter, at one frequency band for all 20 directions. This is regardless of whether one or more types of input features was used (for example, *allAbs* or *allAbs* + *visAbs*). If the input features consider multiple frequency bands, then the output layer will also have multiple values. So a model which has multiband input layer (*allAbs* with width $20 \times 6 = 120$) will have 120 outputs.

Furthermore, it is possible to have models that predict multiple acoustical parameters at the same time, either for single band or multiple bands. In theory, even though this means that the complexity of the model increases as there are more weights that need to be learned, models could benefit from multiple predictions, as all acoustical parameters are related and come from the same physical phenomenon. In practice, no models were tested having an output layer with multiple acoustical parameters. Thus the largest size of the output layer was 120, for multiband models.

4.3.2 Baseline + Location

An additional model was tested, that includes all the possible input layers mentioned earlier, and adds the features indicating the location of both the loudspeakers and the microphone. For example, a model with only all available absorption (*allAbs*) as input and single band, would have input width of $20 \times 1 + 3 + 3 = 26$. Even though the microphone position and orientation is the same for all data examples, this feature was still added to the model as preparation for future work. Moreover, we found that its presence did not affect performance. Unlike the baseline model, when the location features are added, the training and evaluation is done considering the observations for all loudspeaker positions.

The goal of this work is not only to obtain the lowest prediction errors for a specific acoustical parameter, but to explore how different types of input features affect the prediction and to compare the errors across different targets. Therefore, multiple combinations of input and output layers were tested. A summary of some of the modes tested are shown in tables 4.2 and 4.3. Given the vast amount of possible modes, plus their combinations, in this thesis only the most relevant results are presented

Input Layers			
	Frequency	Features	Width
Base.	Single Band	*allAbs*	$1 \times 20 = 20$
		allAbs + *visAbs*	$1 \times 20 + 1 \times 20 = 40$
	Multiband	*allAbs*	$6 \times 20 = 120$
		allAbs + *visAbs*	$6 \times 20 + 6 \times 20 = 240$
Base+Loc	Single Band	*allAbs*	$1 \times 20 + 6 = 26$
		allAbs + *visAbs*	$1 \times 20 + 1 \times 20 + 6 = 46$
	Multiband	*allAbs*	$6 \times 20 + 6 = 126$
		allAbs + *visAbs*	$6 \times 20 + 6 \times 20 + 6 = 246$

Table 4.2: Example of different modes for the input (top) and output (bottom) layers for the baseline and the baseline + location neural network models. Frequency modes can be either single of multiband, while features can select one or more spherical maps *allAbs* , *visAbs* , *nearSurf* or *farSurf* . This table is not exhaustive and shows only a few possible combinations.

Output Layers			
	Frequency	Target	Width
All	Single Band	T_{60}	$1 \times 20 = 20$
		T_{60} + L	$1 \times 20 + 1 \times 20 = 40$
	Multiband	T_{60}	$6 \times 20 = 120$
		T_{60} + L	$6 \times 20 + 6 \times 20 = 240$

Table 4.3: Example of different modes for the output (bottom) layers of some neural network models. Frequency modes can be either single of multiband, while acoustical parameters can select one or more of T_{60}, EDT, C_{50}, EDT or L. This table is not exhaustive and shows only a few possible combinations.

4.3.3 Hyperparameter search

The basic architecture of the baseline model was found with a cursory hyperparameter optimization procedure

An hyperparameter optimization procedure was done to find the best possible architecture of a model for a specific task

In addition to the aforementioned baseline and baseline + location models, a separate set of experiments was run to to study how certain hyperparameters influence the performance of the models, for most input and output modes. Usually, an hyperparameter search (or optimization) procedure is done to find the best possible architecture of a model for a specific task, but on this thesis we are more interested in the analysis rather than the final performance. This procedure was done using

the random search method as defined by [54].

More specifically, we defined both a range of values and a distribution for all valid hyperparameter available in the models. Then, a trial experiment was run, using a randomly sampled hyperparameter set. For each trial, a random sample of the data (using the normalization, split, and augmentation steps mentioned earlier) was used to train and evaluate a model build using that hyperparameter set. Since each trial has an independent random sample of the data and hyperparameter, and each sample follows the same distribution, the trials can be considered as independent experiments of identical distributions (i.i.d) [54]. The main idea is that after enough trials, patterns can be observed in the results, were certain hyperparameter settings consistently perform better than others.

The hyperparameters tested in this process, the range of possible values, and the corresponding probability distribution, are listed in table 4.4. For example, the activation function for each layer is drawn uniformly from the set of 5 possible functions described in the table. On the other hand, the number of layers is drawn geometrically [2] from 1 to 5. For each input and output mode tested, a set of 3000 trials was evaluted.

Group	Hyperparameter	Dist.	Range
Augmentation	size	uniform	[1, 20]
Augmentation	noise σ	uniform	[0, 1]
Model	# layers	geometric	[1, 5]
Model	# hidden units (per layer)	geometric	[5, 100]
Model	activation (per layer)	uniform	{relu, tanh, leakyRelu, clippedRelu, linear}
Optimizer	learning rate	exponential	$[1 \times 10^{-6}, 10]$
Optimizer	gradient threshold	geometric	$[1, 100]$
Regularization	L_2	exponential	$[3.1 \times 10^{-7}, 3.1]$
Regularization	dropout	uniform	$[0, 1]$

Table 4.4: Hyperparameters to be optimized, the range of values that can be selected in the random search, and the distribution used to select their values.

[2] Here we use the same definition for geometrical and exponential drawing as [54]. A geometrical draw from A to B for $0 < A < B$ correspond to a uniform sample in the range $[log(A), log(B)]$, then exponentiating the result to get a number between A and B, and rounding it to the nearest integer. Exponential draw is the same without rounding .

Chapter 5

Results

This chapter starts with a test of the validity of the proposed method by comparing it to the traditional estimation of T_{60} on monophonic (omnidirectional) data. This is followed by an unsupervised analysis of the data, which reveals some characteristics of the data. An expansion of the proposed method to spatial data using 3rd-order hypercardiod signals is then presented. Finally, the role of selected hyperparameters is detailed.

Due to the vast amount of different experiment configurations (determined by loudspeaker position, frequency mode, and acoustical parameter) only the most relevant results are shown on this chapter. Where applicable, additional results can be found in the appendix.

5.1 Test case: T_{60} Estimation vs Sabine Formula

In any machine learning application, it is important to compare the performance of the proposed model to a reference solution. As a proof of concept, the baseline neural method is used to estimate the omnidirectional T_{60} values and the performance is compared to the traditional estimation using the Sabine formula.

For this experiment, the input for the baseline neural network is the all available absorption area spherical map (*allAbs*) for a single band (1 x 20), and the output is a single value (1 x 1), as this is the omnidirectional case. The Sabine formula was computed using the same 20 absorption values. For details about the Sabine formula refer to section 2.1.5.

For each frequency band and speaker position, the data was randomly split into 60 % training and 40 % test sets. The baseline neural network was trained using only the training set. In order to have a fair comparison, the results are only evaluated using the test set. Figure 5.1 shows the prediction accuracy for a typical case, where the neural network model clearly outperforms the Sabine method, with a mean accuracy more than 2 times higher. The Sabine method generally underestimates the T_{60}, and has a wider range of predictions than the true values. For example, the Sabine method predicts a significant amount of values as low as 0.15 s, while the real data never goes under 0.4 s.

Even in cases where the neural network model doesn't perform as well, there is a clear improvement over the Sabine method. For example, for the frequency band of 250 Hz, and the speaker position *farAway*, the Sabine method again has a large bias towards underestimating the values, whereas the prediction error for the neural network model is more even (figure 5.2), with predictions that go over and under the true values on similar ratios.

The boxplots for the distribution of the prediction accuracy of all bands and speaker positions is shown in figure 5.3. For the neural network model, most bands and loudspeaker positions have similar performance, where only the 250 Hz band consistently shows both lower mean and higher variance. For the *farAway* position, the highest frequency also underperformed. On the other hand, the Sabine method shows poor performance along all bands and positions. Interestingly, the best performance happens in the 250 Hz band.

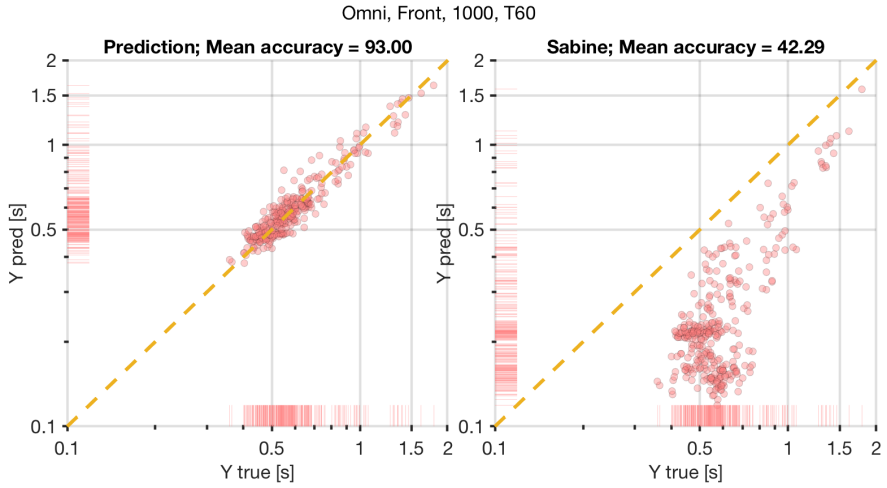


Figure 5.1: Regression plot for the estimation of omnidirectional T_{60} values using the baseline neural network model (left) and the traditional Sabine formula (right), at 1000 Hz, with the speaker position *front*.

5.2 Unsupervised Learning

For each combination of frequency band and acoustical parameter, two dimensionality reduction techniques were applied: PCA and t-SNE. The process consisted of stacking the input features of the all available absorption area spherical map (*allAbs*), with the target values (20 directions for different acoustical parameters), for a total of 40 input features per data sample. Models using all frequency bands (multiband), or all parameters (multiparameter) at the same time, were also considered.

These input features were then projected to a lower dimensionality space utilizing the two methods mentioned previously. As the goal of this projection is to visualize the data and not to reduce the dimensionality of the data as a preprocessing step,

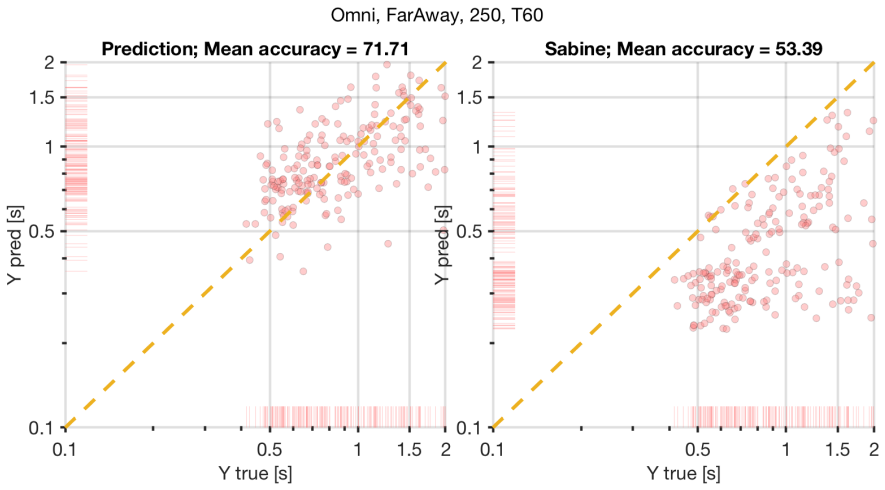


Figure 5.2: Regression plot for the estimation of omnidirectional T_{60} values using the baseline neural network model (left) and the traditional Sabine formula (right), at 250 Hz, with the speaker position *farAway*.

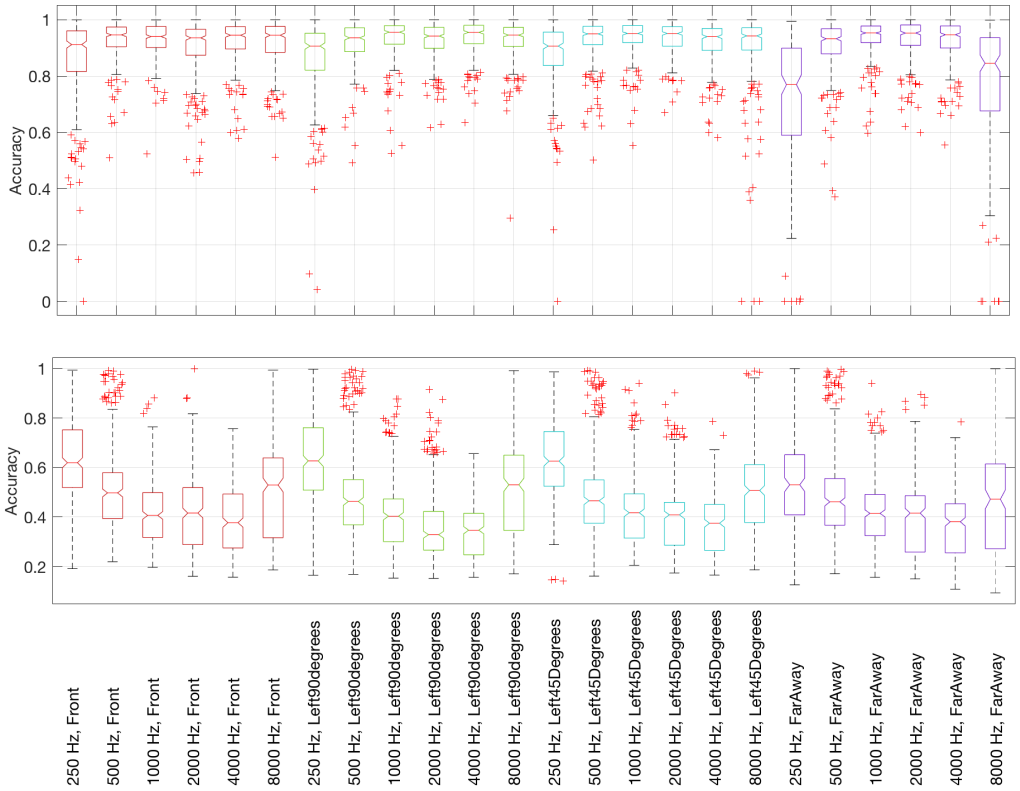


Figure 5.3: Boxplots for the accuracy of the estimation of omnidirectional T_{60} values using the baseline neural network model (top) and the traditional Sabine formula (bottom), at various frequency bands and speaker positions.

all projections were fixed to 2 dimensions. The computation of the mapping and projections was performed using the dimensionality reduction toolbox for Matlab [55].

As there were only 4 loudspeaker positions for all data, it is possible to classify each data sample to one of this distinct positions. All loudspeaker positions were analyzed at the same time, where these class labels were utilized only to visualize the different positions in the projections. The goal was to find out if samples belonging to the same position are clustered close together and separate from other classes in the projection, which would imply that these classes are easily separable giving the selected features.

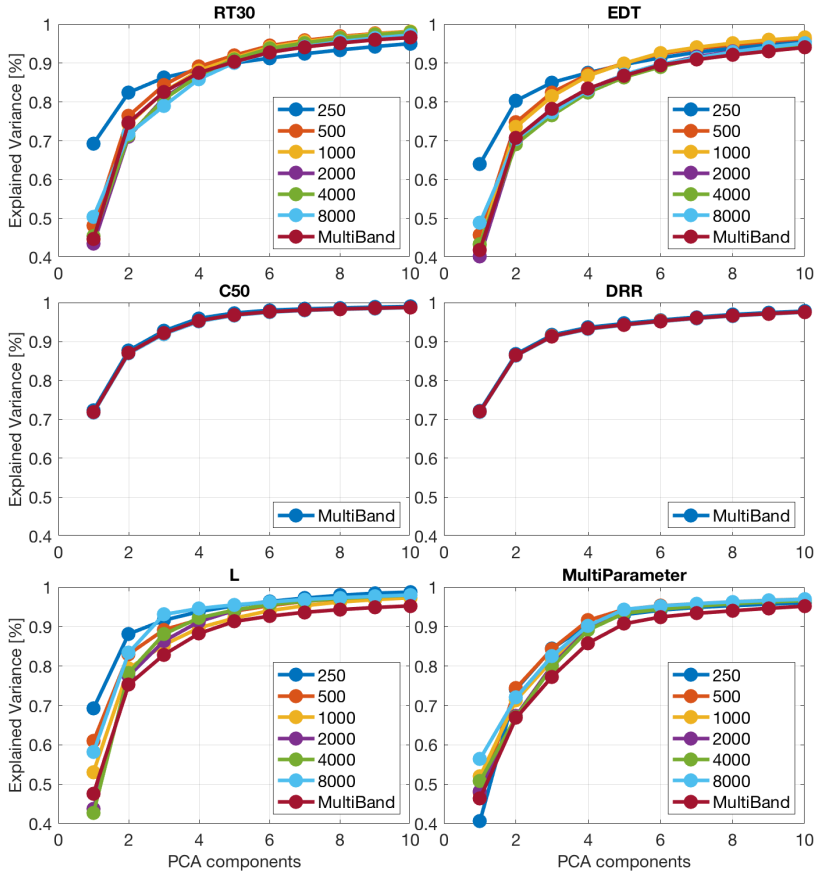


Figure 5.4: Percentage of cumulative explained variance expressed as a function of principal components selected for a low dimensionality projection of various configurations of frequency bands and acoustical parameters.

5.2.1 PCA

By analyzing the eigenvalues obtained form the eigen decomposition of the covariance matrix in the PCA method, it is possible to observe how much variance is explained

by each of the principal component dimensions. Figure 5.4 shows the cumulative explained variance for the first 10 principal components for different acoustical parameters and frequency bands, including the multiband and multiparameter modes. On average, time features such as T_{60} and EDT have under 80% percent of the variance explained using just 2 principal components, while L is slightly better. For all cases, 6 components is already enough to explain at least 90% of the variance, even for the multiband, multiparemeter case.

For some quantities such as T_{60}, the PCA projection shows that the clusters defined by each loudspeaker position are not well separated (see figure 5.5), which means that it is not possible to identify the class based only on T_{60}. However, other parameters such as L show a very clear separation, shown in figure 5.6.

A Trellis plot showing the PCA projections for all frequencies and acoustical parameters is shown in 5.7. There is not much difference in the density of the clusters across different frequency bands, even for the the multiband case where the original dimensionality is much higher.

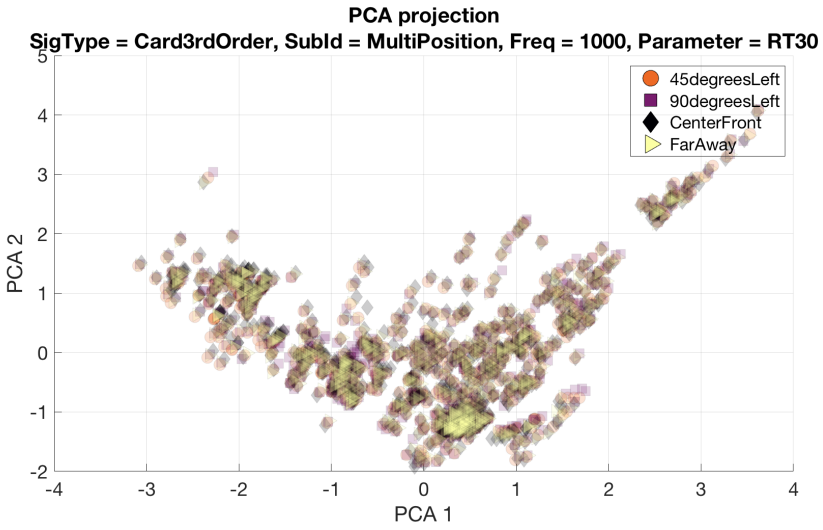


Figure 5.5: Projection into the first 2 principal components using PCA for frequency band of 1000 Hz and acoustical parameter T_{60}.

5.2.2 t-SNE

The t-SNE projections into 2 dimensions are shown in figure 5.8 show little difference to the PCA projections shown earlier. T_{60} and EDT show poor aggregation for loudspeaker positions while other acoustical parameters show good separation between classes. Perhaps the most important distinct feature of these projections is that for some configurations, such as L at 4000 Hz, some of the loudspeaker positions are split into two or more distinct areas, which means that a simple linear classifier would fail to correctly identify at least a sizable portion of the samples.

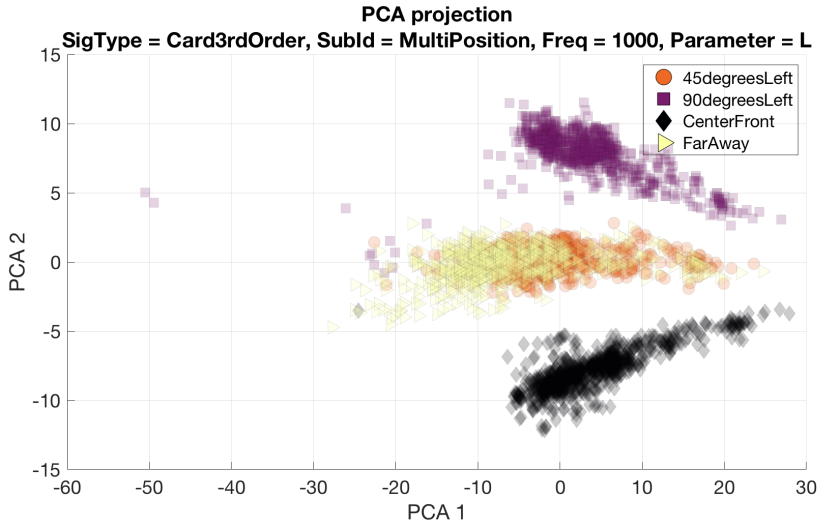


Figure 5.6: Projection into the first 2 principal components using PCA for frequency band of 1000 Hz and acoustical parameter L.

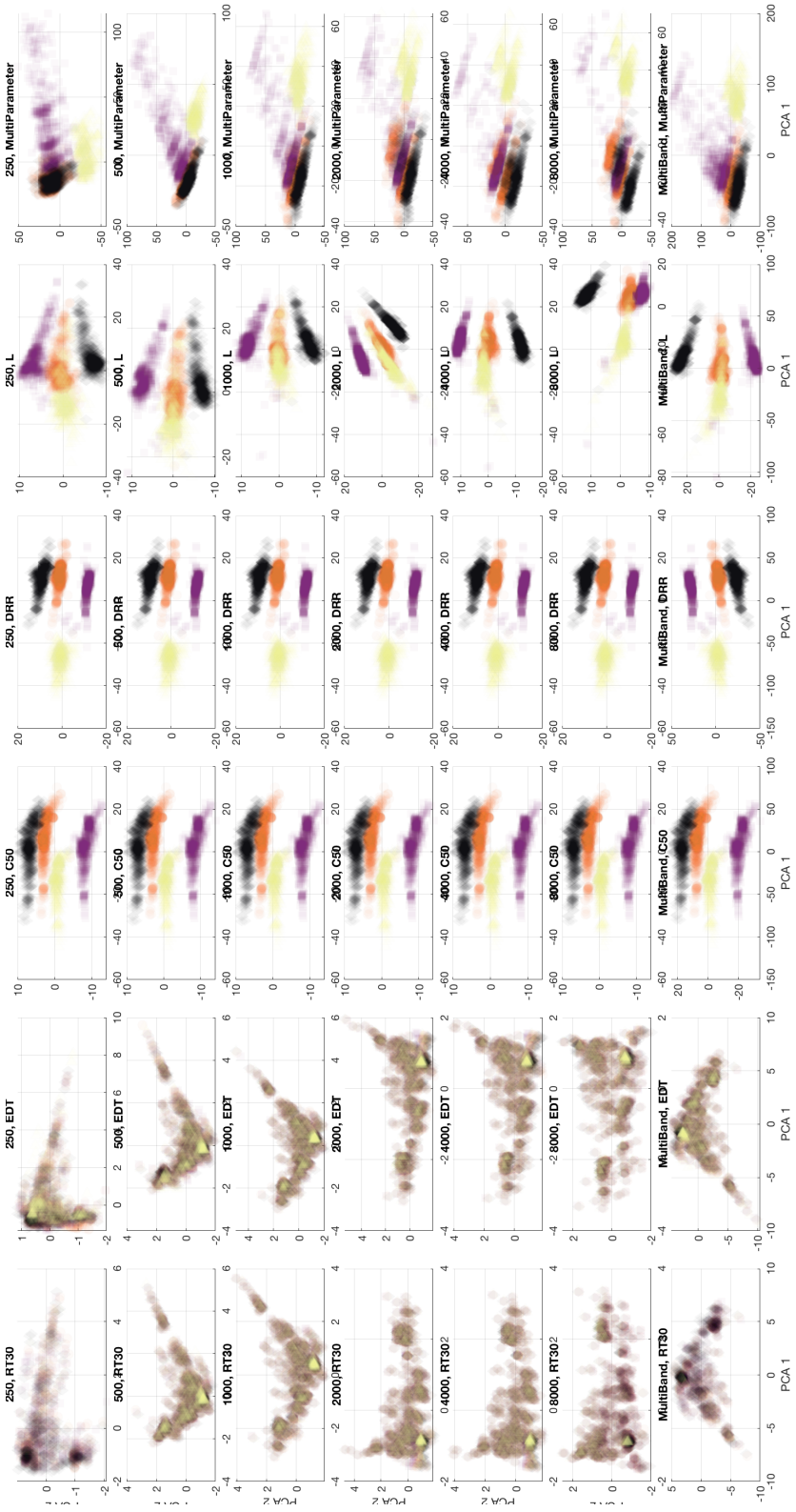


Figure 5.7: Projections into the first 2 principal components using PCA for multiple frequency bands, and acoustical parameters. Plots for C_{50} and DRR are the same for all frequency bands as these values are computed broadband.

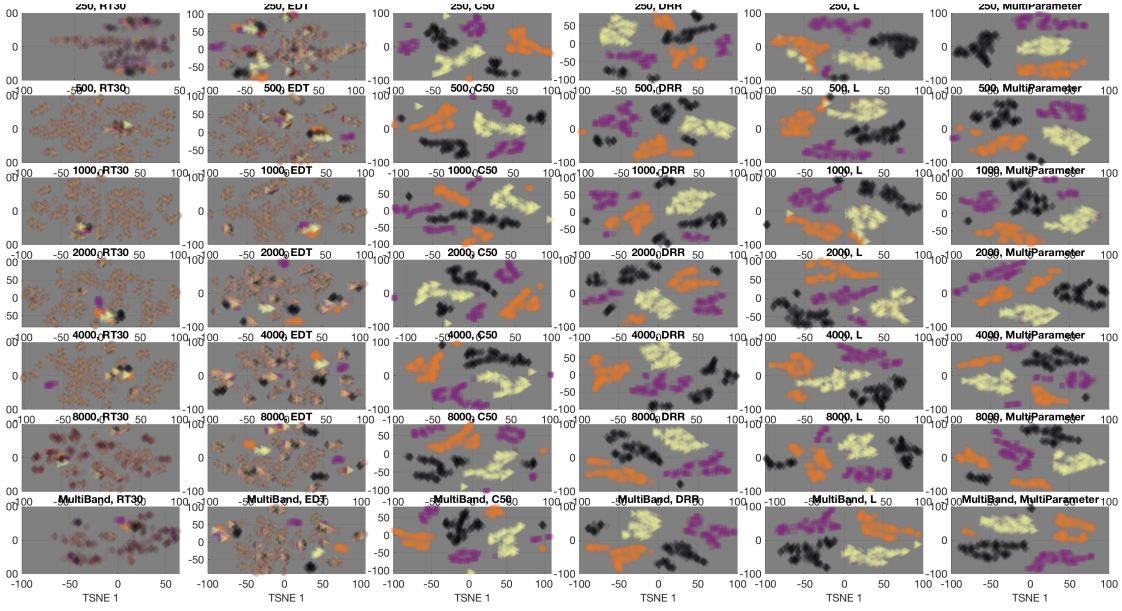


Figure 5.8: Two dimensional projection using t-SNE method for multiple configurations for frequency bands and acoustical parameters. Plots for C_{50} and DRR use the same input features as these values are computed broadband, but the projections correspond to different realizations.

5.3 Estimation

5.3.1 Baseline Model

The baseline neural network model described in chapter 4.3.1, was first trained for each loudspeakeer position individually.

Loudspeaker Location	Frequency	Target	Dimensions	RMSE	MAE	Accuracy
Front	250	T_{60}	20	0.172	0.1093	0.8633
Front	500	T_{60}	20	0.0761	0.0484	0.9251
Front	1000	T_{60}	20	0.0613	0.042	0.9304
Front	2000	T_{60}	20	0.0645	0.0468	0.9245
Front	4000	T_{60}	20	0.052	0.0374	0.9351
Front	8000	T_{60}	20	0.0528	0.0333	0.9329
Front	Multiband	T_{60}	120	0.0943	0.0564	0.9136
Front	250	L	20	1.1011	0.8658	0.9869
Front	500	L	20	1.1213	0.876	0.9862
Front	1000	L	20	1.2575	0.9732	0.9837
Front	2000	L	20	1.2122	0.9133	0.9847
Front	4000	L	20	1.158	0.8605	0.9854
Front	8000	L	20	0.9495	0.6836	0.9878
Front	Multiband	L	120	1.3352	1.0354	0.9828

Table 5.1: Performance results for the baseline neural network model for loudspeaker position *fron*.

5.3.2 Baseline + Location

The Baseline + Location model was trained using the all available absorption spherical map (*allAbs*) for different acoustical parameters using both single and multiband modes. The results are shown in figure 5.10. Overall, this model has the same trend as the baseline, where L has the best performance and DRR the worst. Interestingly there is little difference across frequency bands for most acoustical parameters except for T_{60}.

5.3.3 Hyperparameter Optimization

Impact of Data Size

In this set of experiments, for each configuration of loudspeaker position, frequency mode, and acoustical parameter, the dataset was randomly split into fixed training and validation subsets. Afterwards, a baseline neural network model was trained using only a random sample (with no replacement) of the training subset and evaluated on the validation set. To monitor the impact of the amount of training data, the number of examples in the training sample was varied. To reduce problems derived form

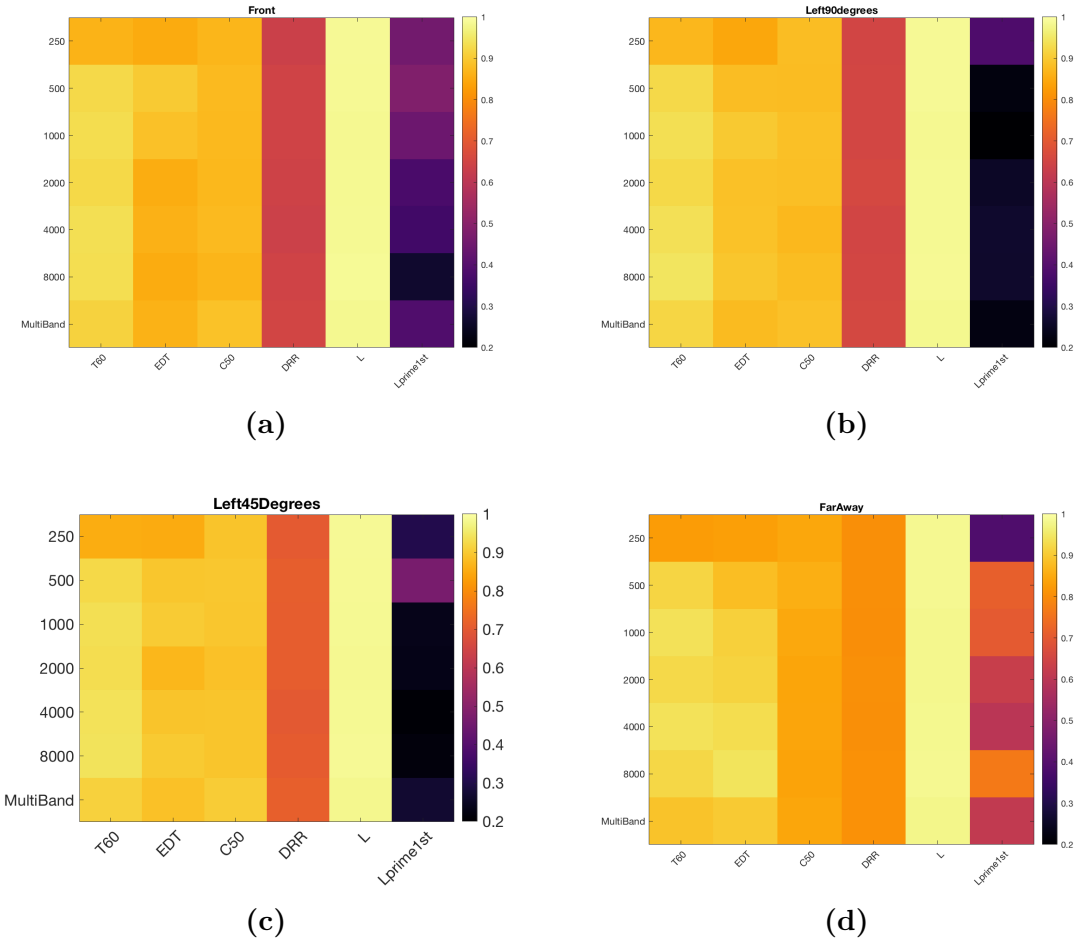
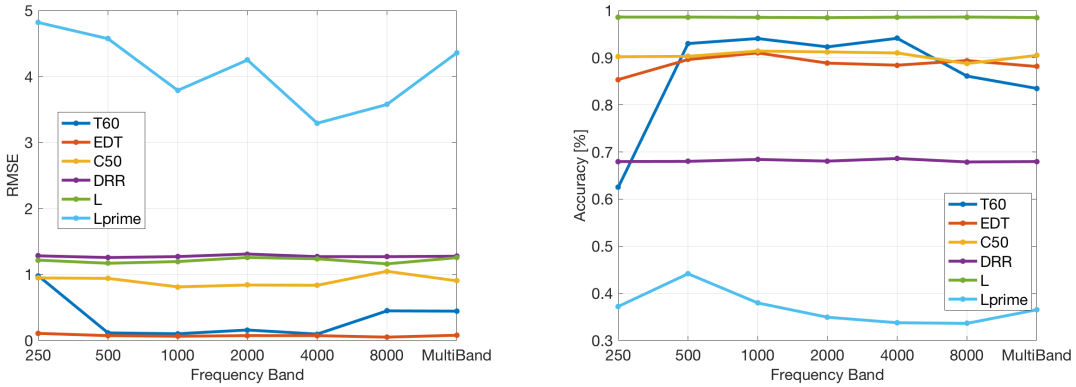


Figure 5.9: Estimation accuracy for the baseline neural network model for all frequency modes and acoustical parameters, for loudspeaker positions *front* (a), *left45degrees* (b), *left90degrees* (c), and *farAway* (d).



(a) **(b)**

Figure 5.10: Performance for the baseline + location neural network model with input features for *allAbs* and the loudspeaker location and orientation coordinates, for all frequency modes and acoustical parameters. (a) shows RMSE where lower is better. (b) shows accuracy where higher is better.

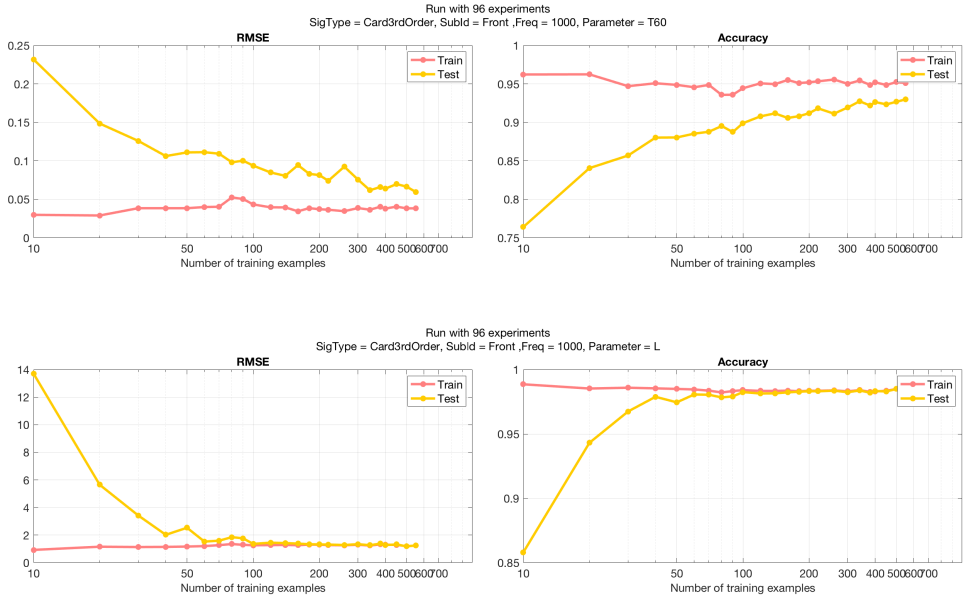


Figure 5.11: Estimation performance for speaker position *front*, frequency 1000 Hz, and acoustical parameter T_{60} (top row) or L (bottom row) when a baseline neural network model is trained on different samples of the training subset. The x-axis is the data size in log-scale and y-axis is the MSE (left column) or estimation accuracy (right column).

having an unlucky sample, each experiment was repeated 4 times, and the average performance of all repetitions was reported. With 5 speaker positions (4 single speaker and 1 multispeaker), 7 frequency modes (6 single band and 1 multiband), 6 acoustical parameters, 25 training subset sizes, and 4 repetitions, a total of 21 000 experiments were evaluated. All other hyperparameters remained constant.

Figure 5.11 shows the performance of a typical configuration for the acoustical parameters of T_{60} and L for increasing training sample sizes. At first, the results follow the expected behavior, were adding more data increases performance. However, the performance saturates after around 150 examples but at a very high accuracy value. This means that at least for this speaker position and frequency band, the estimation problem is quite simple, and the difference between predictions and true values could very well be the product of measurement noise or deficiencies on the computation of the T_{60} from the recorded impulse responses. Moreover, this performance is present in both training and validation subsets, therefore showing little sign of overfitting.

The results for other speaker locations shown in figure 5.12 resemble the previous ones, where only loudspeaker position *farAway* has a significantly diminished performance.

Other configurations show similar behavior. Once again, for single band models, frequencies of 250 and 8000 Hz show overall lower performance but the trend is the same. Figure 5.13 shows a comparison of the same curves as 5.11 across multiple frequency modes.

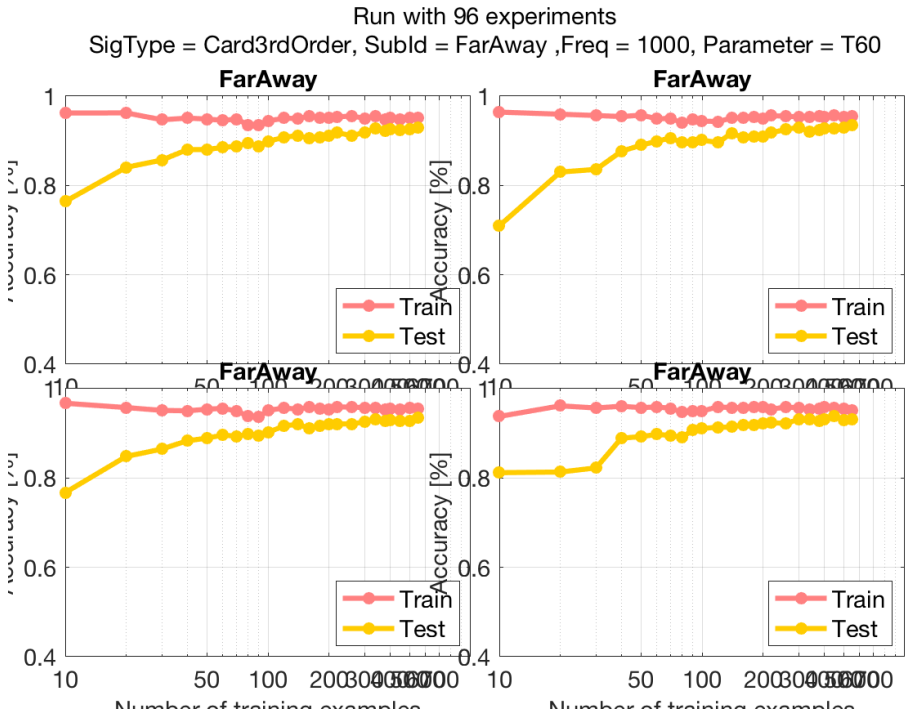


Figure 5.12: Estimation performance for frequency 1000 Hz, and acoustical parameter T_{60} for loudspeaker positions *front* (top left), *left90degrees* (top right), *left45degrees* (bottom left) and *farAway* (bottom right) when a baseline neural network model is trained on different samples of the training subset. The x-axis is the data size in log-scale and y-axis is the estimation accuracy.

Impact of Model Capacity

To study the relationship between model capacity and performance the results of all individual experiments of the random search for a single configuration were sorted according to the total number of trainable parameters present in the model. All the models used are fully connected neural networks, so the total amount of these parameters is determined by the number of layers and the number of hidden units per layer. Here there is no difference between deep (many layers) and wide (many units per layers) models, as the objective is to measure the overall capacity of the model.

Figure 5.14 shows the relationship between accuracy and model capacity for randomized experiments. There is no clear dependency between the number of the trainables and the overall accuracy. Models with as few as 225 trainable parameters (which is a single hidden layer with 5 hidden units) can already achieve very high accuracy. More surprisingly, there seems to be two concentration of results, where most of the results are either good enough (with accuracy larger than 80%), or very poor (with accuracy lower than 10%, and only a few amount of results are in between. This means that the problem is simple enough that it can be solved by a vast amount of model architectures, as long as certain features are present.

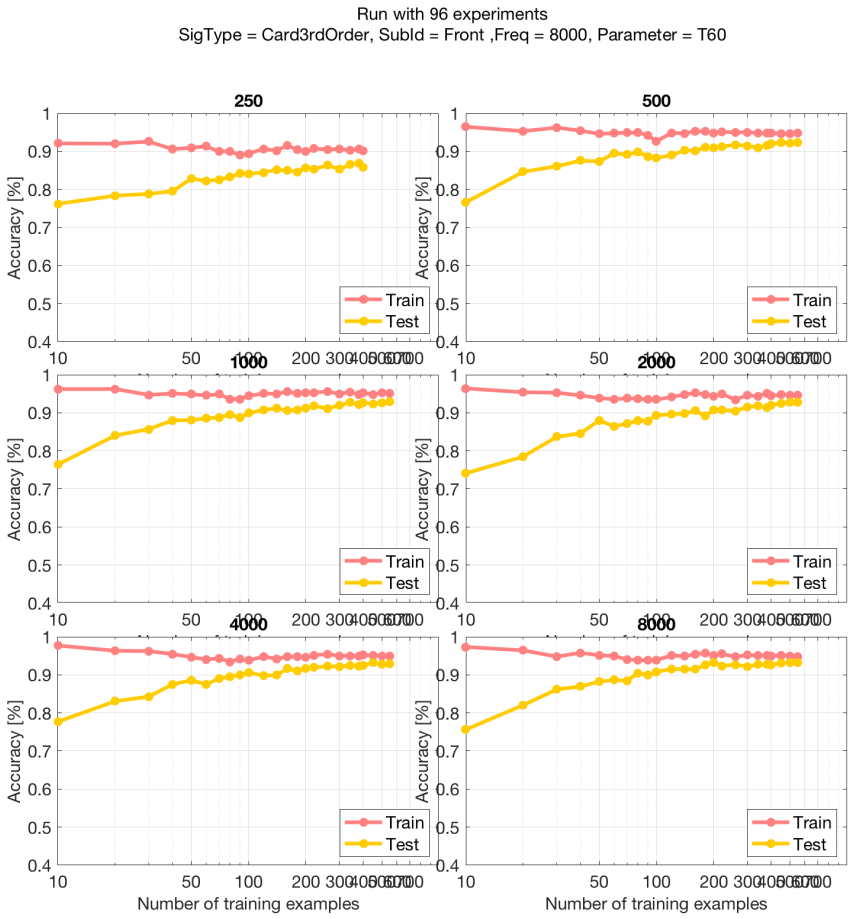


Figure 5.13: Comparison of the estimation performance loudspeaker positions *front* acoustical parameter T_{60} for multiple frequency bands when a baseline neural network model is trained on different samples of the training subset. The x-axis is the data size in log-scale and y-axis is the estimation accuracy.

The same pattern is present in models with a larger amount of inputs and outputs. For example, models that predict the values for all loudspeaker positions at the same time are shown in figure 5.15, while models that predict all loudspeaker positions and all frequency modes appear in figure 5.16.

Impact of Model Depth

The total number of trainable parameters is not the only way to measure a model's complexity, as the ordering composition and distribution of those parameters along the full architecture is important. To study the relationship between model depth (number of layers) and performance the results of all experiments for a given configuration were grouped by depth.

Generally, deeper and thin models are better than shallow and wide ones [56], [7], however, here shallow models perform slightly better. Figure shows the 5.17 the

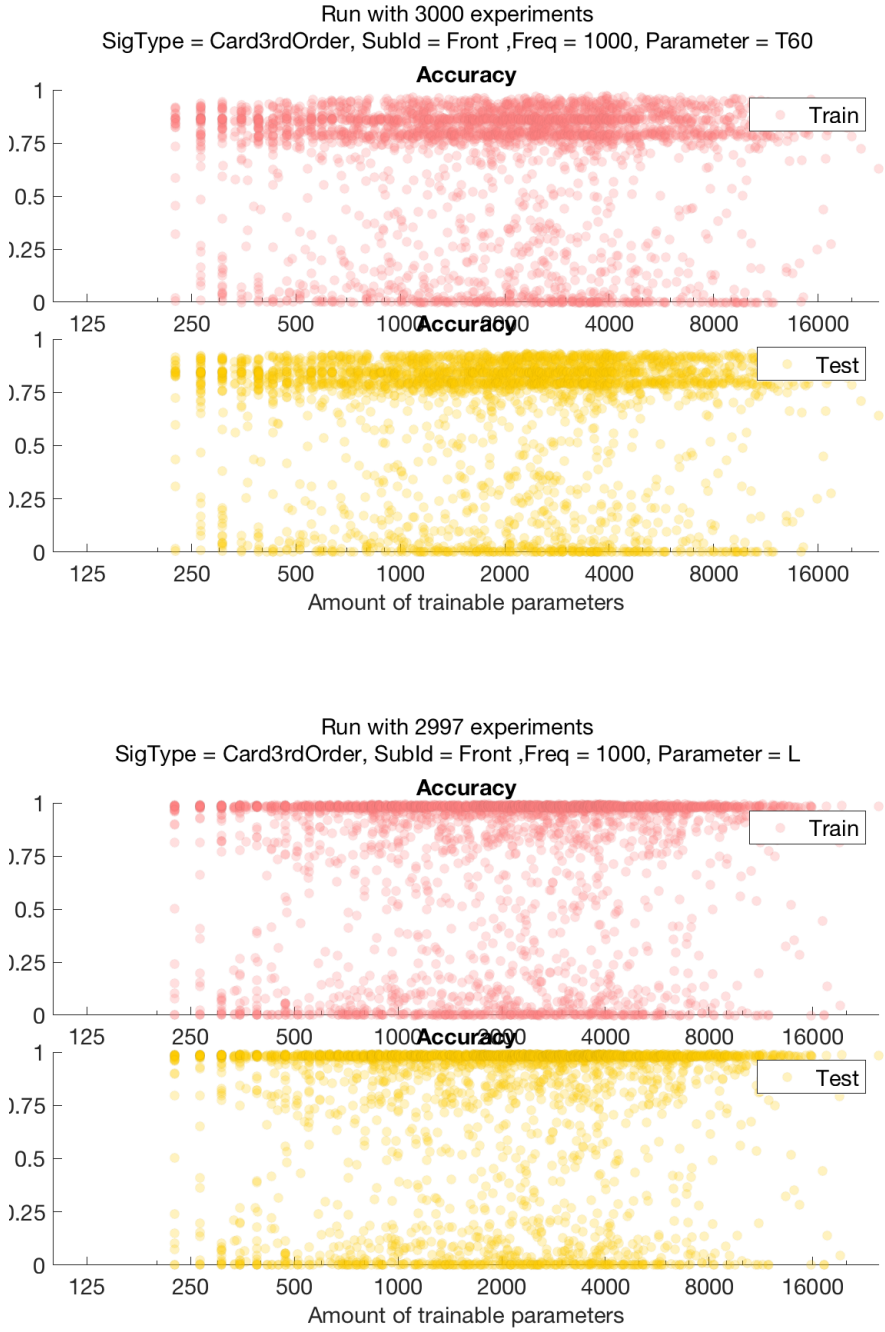


Figure 5.14: Estimation performance for speaker position *front*, frequency 1000 Hz, and acoustical parameter T_{60} (top two rows) or L (bottom two row) when training different neural networks models. The x-axis is total amount of trainable parameters in the model and y-axis is the estimation accuracy for train subset (rows 1 and 3) and test subset (rows 2 and 4).

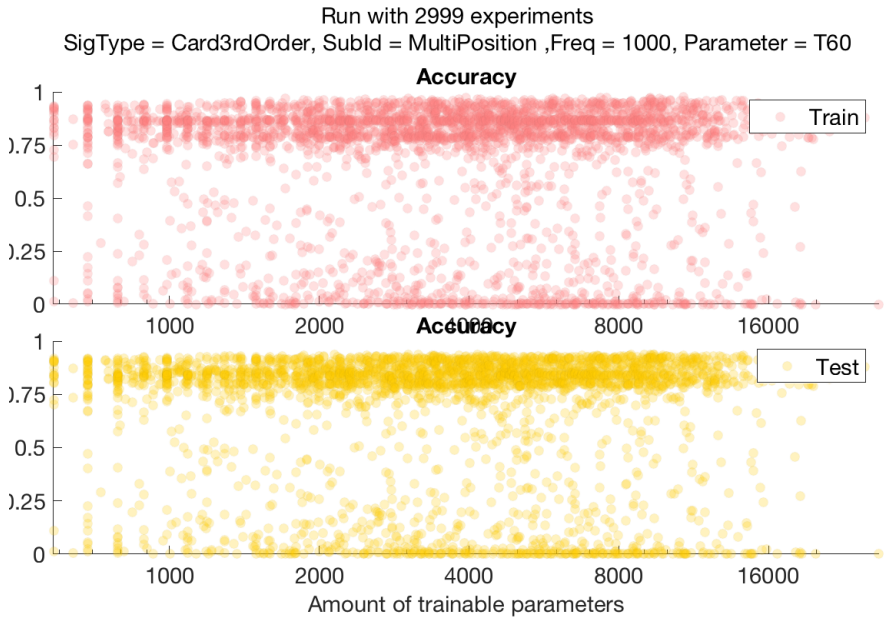


Figure 5.15: Estimation performance for all speaker positions, frequency 1000 Hz, and acoustical parameter T_{60} when training different neural networks models. The x-axis is total amount of trainable parameters in the model and y-axis is the estimation accuracy for train subset (top row and test subset (bottom row).

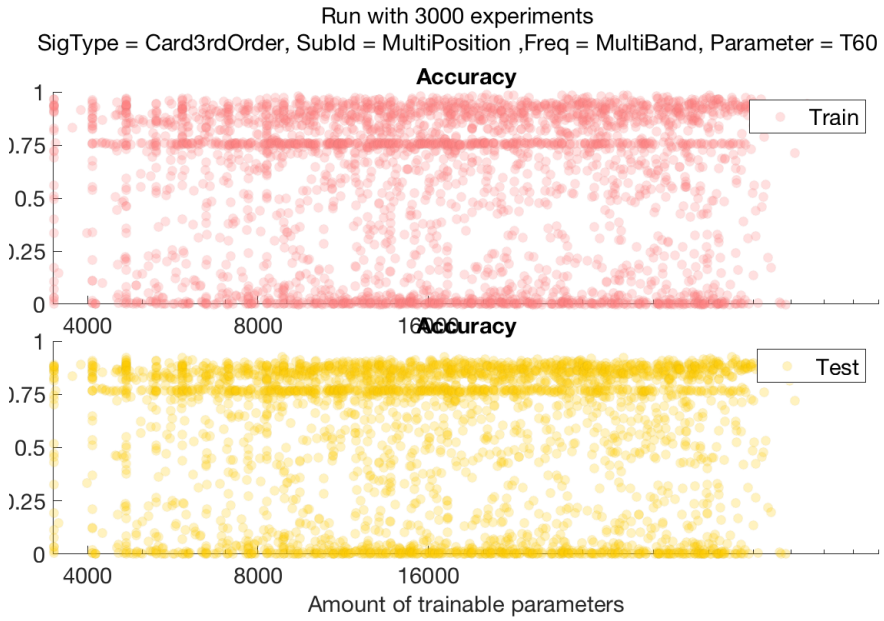


Figure 5.16: Estimation performance for all speaker positions, all frequency bands, and acoustical parameter T_{60} when training different neural networks models. The x-axis is total amount of trainable parameters in the model and y-axis is the estimation accuracy for train subset (top row and test subset (bottom row).

boxplots for the accuracy values of all trained models grouped by number of layers for some configurations. Other configurations followed a similar trend.

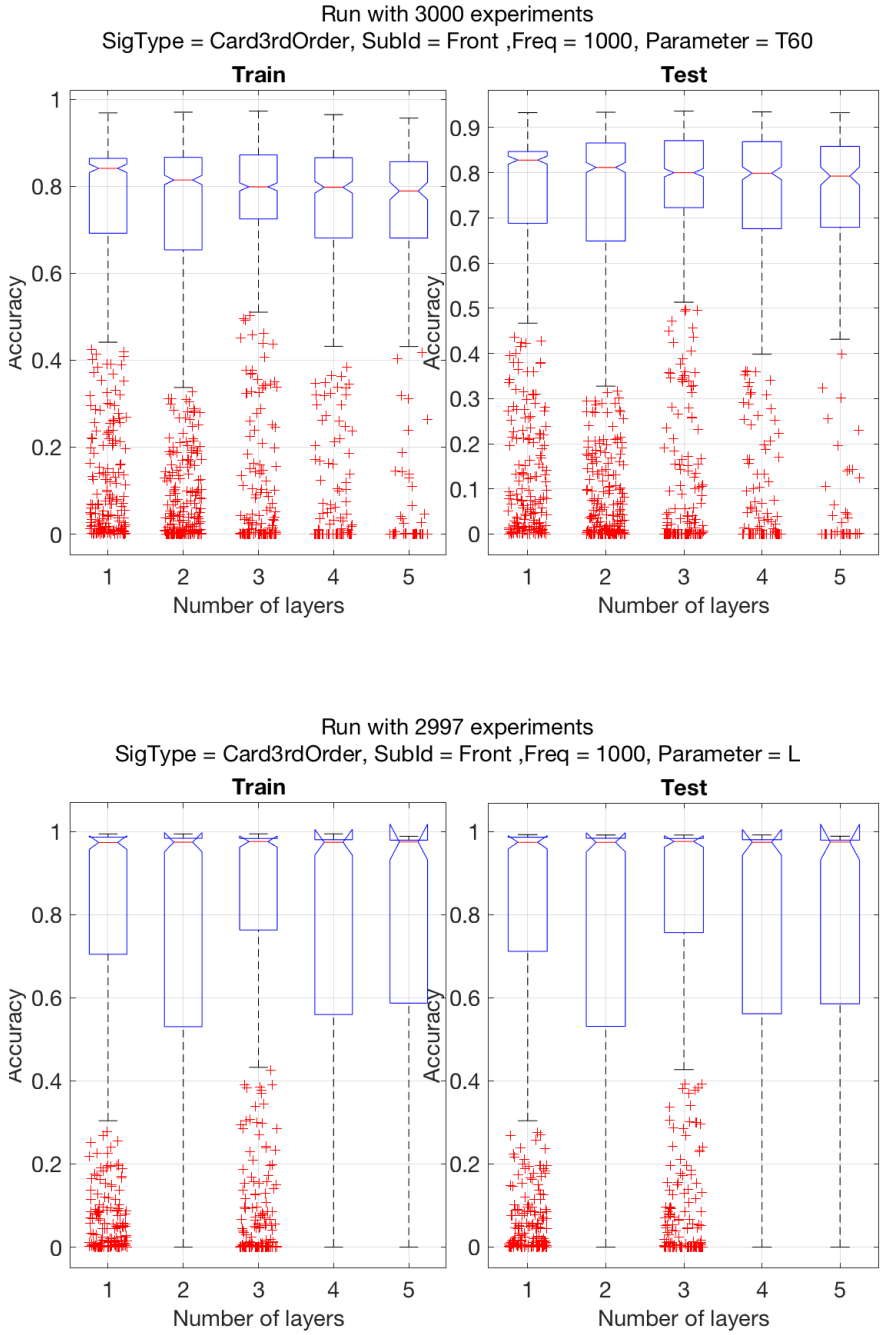


Figure 5.17: Boxplots for the estimation performance of 3000 different models with randomized hyperparameters grouped by number of layers, for loudspeaker position *front*, frequency 1000 Hz, and acoustical parameters T_{60} (top) and L (bottom). The x-axis is the number of hidden layers and the y-axis is the estimation accuracy.

Chapter 6

Discussion

6.1 Results Analysis

The exploratory data analysis of the room acoustics dataset confirmed that some acoustical parameters like T_{60} have little spatial variety, while others like L are more directional. Additionally, there is some correlation between some of the measured values and the total amount of absorption area in the room.

The boxplots for the values computed for T_{60} show a clear gap, where there is a large group of results under 1 s and a smaller group above 1 s. This could mean that all values over 1 s are incorrect, derived from errors or bugs in the software used to compute the values.

The baseline model is much better than the traditional Sabine method, and it can be expanded to estimate other acoustical parameters as well. That said, not all parameters show the same performance. In general, T_{60}, EDT and L are easy to estimate for most frequency bands, while C_{50} and DRR are more challenging. Although this could be because the latter ones are computed broadband.

Perhaps the most surprising result is that there is little sensitivity to most hyperparameters for the models, where many different neural network architectures perform well. In machine learning applications, the hyperparameter optimization is usually a crucial step, where finding the correct configuration will determine performance. In this case, the physical model of sound propagation appears to be simple enough that it can be approximated by a vast range of functions, expressed as networks with many architectures. However, this needs to be analyzed further, as it could only be true for the particular dataset used, which is limited to a single room, and fixed positions for microphone and loudspeakers.

More importantly, models with very few trainable parameters can achieve high performance. These models could then be utilized in real time implementations due to their reduced computational cost. That said, it is important to also consider the computational cost of the data pre-preprocessing step described in section 3.2.1, as the input features (the spherical maps using absorption coefficients) need to be computed for each microphone position.

The baseline model with location features also performed very well. Furthermore, the results of section 5.12 where performance is validated against amount of training

examples show that relatively low amount of data is needed, at least for the fixed microphone position. In practice, it might be that a simple room as the once used requires only a few dozen measurements to estimate the effects of all possible loudspeaker positions. More complex conditions might require more measurements.

6.2 Future Work

The work contained in this thesis provides a proof of concept on how to apply machine learning methods to estimate common room acoustics parameters based on real physical measurements. The results so far are promising and show that the method works very well for the conditions available. However, these results are only the first step of a much bigger problem, and there are several other things that need to be tried.

The most obvious next step is to add more, and more diverse data. The dataset used was restricted to measurements inside a single room with a simple rectangular shape, limited positions for the microphone and loudspeakers, and a relatively small number of configurations for objects inside the room. Rooms with more complex, asymmetrical geometries might present different behavior, as the sound propagation becomes more complex in such situations. Furthermore, the results show that some loudspeaker positions are more difficult than others, specially those were the loudspeaker is very close to a wall, and not pointing towards the microphone. Additional complexity can be added by introducing diffusing elements to the room. An ideal dataset would include data from many different rooms of multiple sizes, geometries, acoustical materials, and plenty of measurements with multiple non fixed positions of microphones and loudspeakers covering a large area of each room.

Obtaining real life data is difficult and costly. An alternative could be using data from simulations. In theory, these simulations could provide infinite data where the precision is only constrained by computational power. An interesting topic would be to compare the performance of models using simulated data against real world data. Moreover, a transfer learning approach could be useful, where the models are trained using mostly simulated data, and then refined using limited real life data.

Finding a good representation for different room shapes could be a problem, as the current models lack input features for this property, and adding features for the dimensions of walls will not capture nuances in the geometry, such as angled surfaces. An idea to test is to use image source model to generate a grid of points that describes the room geometry.

Other types of input features could also be explored. In this work only a fraction of the available features were tested, mostly the total absorption present for all particular directions. Information such as visible absorption and distance to surfaces were not fully utilized. Of particular interest are visual features such as spherical (or panoramic) photographs of the rooms, as these images are easy to collect and don't require as much effort as 3d modelling. Additionally, models that employ different types or features (such as visual, geometrical or absorption coefficients) will need to carefully normalize all features, or perhaps use an ensemble model approach.

Finally, a more ambitious project should explore if it is possible to generate the full room impulse response (for as many channels or directions as needed) using the available features or additional ones. That said, given the length of a typical impulse response (up to a few seconds), a generative model would probably require much more data than the currently available.

Chapter 7

Conclusion

This thesis has provided a proof of concept for a machine learning method to estimate a set of typical room acoustics parameters using only geometric information as input features. All models were trained and evaluated in a dataset that contains real world acoustical measurements, as well as various geometric information including absorption coefficients α. The process included a thorough analysis of the dataset, along with a signal processing step where microphone array encoding methods are used to extract spatial information of the data.

From this analysis, we confirm that reverberation times, measured by both T_{60} and EDT, have little variance across different directions for most examples. On the other hand, parameters such as L, that depend more on the direct sound than the late reflections, show a clear correlation with the absorption area available on each direction. Additionally, a dimensionality reduction for visualization purposes showed a similar trend. When clustering by loudspeaker position, T_{60} and EDT have little cluster separation, while L, DRR, and C_{50} show good separation.

The estimation was performed using neural network models. A basic baseline model using a FC network and the spherical map of available absorption as input features, achieved good results for most parameters. The model was first validated against the Sabine method, which the proposed model consistently outperforms. An additional model that includes input features for the locations of the source and microphone, also achieved high performance.

Furthermore, the hyperparameter analysis of both models produced three main findings. First, a large number of combinations of hyperparameters produce very high performance, where even models with as few a 1 layer and 15 hidden units can match the results of models with thousands of trainable parameters. Second, the depth of the models has little influence on the results, where there is only a slight improvement for models up to 2 layers. Third, the benefit of increasing the amount of training data examples for a single loudspeaker saturates after around 100 examples. All these suggest that the acoustical parameters derived from the sound propagation inside the analyzed room, can be easily modeled by a wide range of neural networks architectures.

Bibliography

[1] A. F. Jarosz, G. J. Colflesh, and J. Wiley, "Uncorking the muse: Alcohol intoxication facilitates creative problem solving," *Consciousness and Cognition*, vol. 21, no. 1, pp. 487 – 493, 2012. Beyond the Comparator Model.

[2] T. Standage, *A history of the world in 6 glasses*. Walker & Co., 1 ed., 2005.

[3] M. Schroeder, T. D. Rossing, F. Dunn, W. M. Hartmann, D. M. Campbell, and N. H. Fletcher, *Springer Handbook of Acoustics*. Springer Publishing Company, Incorporated, 1st ed., 2007.

[4] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," *The Journal of the Acoustical Society of America*, vol. 138, no. 2, pp. 708–730, 2015.

[5] K. Kowalczyk and M. van Walstijn, "Room acoustics simulation using 3-d compact explicit fdtd schemes," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 34–46, Jan 2011.

[6] R. Mehra, N. Raghuvanshi, L. Antani, A. Chandak, S. Curtis, and D. Manocha, "Wave-based sound propagation in large open scenes using an equivalent source formulation," *ACM Trans. Graph.*, vol. 32, pp. 19:1–19:13, Apr. 2013.

[7] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 6231–6239, Curran Associates, Inc., 2017.

[8] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era.," in *ICCV*, pp. 843–852, IEEE Computer Society, 2017.

[9] R. Ratnam, D. L. Jones, B. C. Wheeler, W. D. O'Brien, C. R. Lansing, and A. S. Feng, "Blind estimation of reverberation time," *The Journal of the Acoustical Society of America*, vol. 114, no. 5, pp. 2877–2892, 2003.

[10] O. A. B. Hassan, *Building Acoustics and Vibrations: Theory and Practice*. World Scientific, 2009.

[11] A. Cops, J. Vanhaecht, and K. Leppens, "Sound absorption in a reverberation room: Causes of discrepancies on measurement results," *Applied Acoustics*, vol. 46, no. 3, pp. 215 – 232, 1995. Building Acoustics.

[12] P. Laukkanen, "Evaluation of studio control room acoustics with spatial impulse responses and auralization," Master's thesis, AAlto University, 2014.

[13] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill School Education Group, 2nd ed., 2001.

[14] G. K. Behler, "How to Compare Concert Halls by Listening to Music." http://acoustics.org/pressroom/httpdocs/152nd/behler.html, accesed on September 2018.

[15] I. 3382, *Acoustics - Measurement of the reverbration time of rooms with reference to other acoutics parameters*. International Standards Organisation, 1997.

[16] L. L. Beranek, "The notebooks of wallace c. sabine," in *The Journal of Acoustical Socierty of America*, pp. 629–639, 1977.

[17] V. Pulkki, *Communication Acoustics: An Introduction to Speech, Audio, and Psychoacoustics*. Chichester, West Sussex, United Kingdom: Wiley, 1 ed., 2015.

[18] C. Hak, R. Wenmaekers, and L. Luxemburg, van, "Measuring room impulse responses : impact of the decay range on derived room acoustic parameters," *Acta Acustica united with Acustica*, vol. 98, no. 6, pp. 907–915, 2012.

[19] P. Zahorik, "Direct-to-reverberant energy ratio sensitivity," *The Journal of the Acoustical Society of America*, vol. 112, no. 5, pp. 2110–2117, 2002.

[20] S. Goetze, E. Albertin, M. Kallinger, A. Mertins, and K. Kammeyer, "Quality assessment for listening-room compensation algorithms," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2450–2453, March 2010.

[21] L. McCormack, S. Delkaris-Manias, and V. Pulkki, "Parametric acoustic camera for real-time sound capture, analysis and tracking," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17), Edinburgh, UK*, pp. 5–9, 2017.

[22] D. Pavlidi, S. Delikaris-Manias, V. Pulkki, and A. Mouchtaris, "3d doa estimation of multiple sound sources based on spatially constrained beamforming driven by intensity vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016*, vol. 2016-May of *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (United States), pp. 96–100, Institute of Electrical and Electronics Engineers, 5 2016.

[23] V. Pulkki, S. Delikaris-Manias, and A. Politis, *Parametric Time-Frequency Domain Spatial Audio*. Wiley, 1st ed., 2017.

[24] B. Rafaely, *Fundamentals of Spherical Array Processing*. Springer-Verlag Berlin Heidelberg, 1 ed., 2015.

[25] L. McCormack, S. Delikaris-Manias, A. Farina, D. Pinardi, and V. Pulkki, "Real-time conversion of sensor array signals into spherical harmonic signals with applications to spatially localized sub-band sound-field analysis," in *Audio Engineering Society Convention 144*, May 2018.

[26] S. Moreau, J. Daniel, and S. Bertet, "3D sound field recording with higher order ambisonics Objective measurements and validation of a 4th order spherical microphone," in *120th AES Convention*, (Paris, France), Audio Engineering Society (AES), 2006.

[27] D. Piretzidis, "Surface Spherical Harmonic Functions Visualization." https://ww2.mathworks.cn/matlabcentral/fileexchange/44869-surface-spherical-harmonic-functions-visualization?s_tid=prof_contriblnk, accessed on September 2018.

[28] S. Delikaris-Manias, D. Pavlidiy, V. Pulkki, and A. Mouchtaris, "3d localization of multiple audio sources utilizing 2d doa histograms," in *Proceedings of the 24th European Signal Processing Conference, EUSIPCO 2016*, vol. 2016-November of *European Signal Processing Conference*, (United States), pp. 1473–1477, Institute of Electrical and Electronics Engineers, 11 2016.

[29] A. Politis, "Microphone array processing for parametric spatial audio techniques," 2016.

[30] A. Politis and H. Gamper, "Comparing modeled and measurement-based spherical harmonic encoding filters for spherical microphone arrays," in *WASPAA*, pp. 224–228, IEEE, 2017.

[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[32] S. Rogers and M. Girolami, *A First Course in Machine Learning, Second Edition*. Chapman & Hall/CRC, 2nd ed., 2016.

[33] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 3rd ed., 2014.

[34] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative review," *J Mach Learn Res*, vol. 10, pp. 66–71, 2009.

[35] L. van der Maaten, "Learning a parametric embedding by preserving local structure," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics* (D. van Dyk and M. Welling, eds.), vol. 5 of *Proceedings of Machine Learning Research*, (Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA), pp. 384–391, PMLR, 16–18 Apr 2009.

[36] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," 2008.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization.," *CoRR*, vol. abs/1412.6980, 2014.

[38] G. E. Buyo, "ML4U Learn to model AI: Capacity, Overfitting and Underfitting." http://ml4u.net/introduction/capacity_overfitting_underfitting/, accessed on September 2018.

[39] A. NG, "Machine Learning Yearning." http://www.mlyearning.org, accessed on September 2018. Preprint.

[40] A. Y. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data.," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[42] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[43] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "Fma: A dataset for music analysis," 2016. cite arxiv:1612.01840Comment: ISMIR 2017 camera-ready.

[44] R. Stewart and M. Sandler, "Database of omnidirectional and b-format impulse responses," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, March 2010.

[45] S. Adavanne, A. Politis, and T. Virtanen, "TUT Sound Events 2018 - Ambisonic, Reverberant and Synthetic Impulse Response Dataset," Apr. 2018.

[46] S. Adavanne, J. Nikunen, A. Politis, and T. Virtanen, "TUT Sound Events 2018 - Ambisonic, Reverberant and Real-life Impulse Response Dataset," Apr. 2018.

[47] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, pp. 5220–5224, IEEE, 2017.

[48] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*, Feb 2000.

[49] J. Baez, "Who Discovered the Icosahedron?." http://math.ucr.edu/home/baez/icosahedron/, accesed on September 2018.

[50] Institute of Sound Recording, "IoSR Matlab Toolbox." https://github.com/IoSR-Surrey/MatlabToolbox, accessed on May 2018.

[51] V. Cox, "Exploratory data analysis," in *Translating Statistics to Make Decisions*, Berkeley, CA: Apress, 2017.

[52] D. van den Assem, "Predicting periodic and chaotic signals using wavenets," Master's thesis, Delt University of Technology, 2017.

[53] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679 – 688, 2006.

[54] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[55] L. van der Maaten, "Matlab Toolbox for Dimensionality Reduction." https://lvdmaaten.github.io/drtoolbox/, accesed on September 2018.

[56] H. Mhaskar, Q. Liao, and T. A. Poggio, "When and why are deep networks better than shallow ones?," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 2343–2349, 2017.

[57] M. Lovedee-Turner and D. Murphy, "Dataset for: Application of Machine Learning for the Spatial Analysis of Binaural Room Impulse Responses," Oct. 2017. Funding was provided by a UK Engineering and Physical Sciences Research Council (EPSRC) Doctoral Training Award, the Department of Electronic Engineering at the University of York.