# Intelligent Control of Dynamic Range Compressor

Di Sheng

Submitted in partial fulfillment of the requirements of the
Degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London

2019

# Statement of Originality

I, Di Sheng, confirm that the research included within this thesis is my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.



Signature:

Date: 06/09/2019

**Abstract**

Music production is an essential element in the value chain of modern music. It includes enhancing the recorded audio tracks, balancing the loudness level of multiple tracks as well as making artistic decisions to satisfy music genre, style and emotion. Similarly to related professions in creative media production, the tools for music making are now highly computerised. However, many parts of the work remain labour intensive and time consuming. The demand for intelligent tools is therefore growing. This situation encourages the emerging trend of ever increasing research into intelligent music production tools. Since audio effects are among the main tools used by music producers, there are many discussions and developments targeting the controlling mechanism of audio effects. This thesis is aiming at pushing the boundaries in this field by investigating the intelligent control of one of the essential audio effects, the dynamic range compressor.

This research presents an innovative control system design. The core of this design is to learn from a reference audio, and control the dynamic range compressor to make the processed input audio sounds as close as possible to the reference. One of the proposed approaches can be divided into three stages, a feature extractor, a trained regression model, and an objective evaluation algorithm. In the feature extractor stage we firstly test feature sets using conventional audio features commonly used in speech and audio signal analyses. Substantially, we test handcrafted audio features specifically designed to characterise audio properties related to the dynamic range of audio samples. Research into feature design has been completed at different levels of complexity. A series of feature selection schemes are also assessed to select the optimal feature sets from both conventional and specifically designed audio features. In the subsequent stage of the research, feature extraction is replaced by a feature learning deep neural network (DNN). This is addressing the problem that the previous features are exclusive to each parameter, while a general feature extractor may be formed using DNN. A universal feature extractor can reduce the computational cost and become easier to adapt to more complex audio materials as well. The second stage of the control system is a trained regression model. Random forest regression is selected from several algorithms using experimental validation. Since different feature extractors are tested with increasingly complex audio material, as well as exclusive to the DRC's parameters, e.g., attack time or compression ratio, separate models are trained and

tested respectively. The third component of our approach is a method for evaluation. A computational audio similarity algorithm was designed to verify the results using auditory models. This algorithm is based on estimating the distance between two statistical models fitted on perceptually motivated audio features characterising similarity in loudness and timbre. Finally, the overall system is evaluated with both objective and subjective methods.

The main contribution of this Thesis is a method for using a reference audio to control a dynamic range compressor. Besides the system design, the analysis of the evaluation provides useful insights of the relations between audio effects and audio features as well as auditory perception. The research is conducted in a way that it is possible to transfer the knowledge to other audio effects and other use case scenarios, providing an alternative research direction in the field of intelligent music production and simplifying how audio effects are controlled for end users.

To dream the impossible dream,

To fight the unbeatable foe,

To bear with unbearable sorrow,

To run where the brave dare not go.

To right the unrightable wrong,

To love pure and chaste from afar,

To try when your arms are too weary,

To reach the unreachable star.

---

*Man of La Mancha*

*Don Quixote*

*"The Impossible Dream"*

# Acknowledgements

Writing this thesis is like walking through the PhD journey again. The support I received along the way is as vivid as yesterday. This thesis would not be possible without the help along the way.

First of all, I would like to thanks my supervisor, György Fazekas, for his guidance and encouragement. He has generously invested so much time and effort on supervising my PhD process. His knowledge and experience has supported me to overcome many difficulties.

I would also like to thank Prof. Mark Sandler and Prof. Josh Reiss who were on my research panel and examined my stage milestones. My gratitude also goes to the Center of Digital Music (C4DM), Queen Mary University of London for funding this PhD research. The current and the previous members of the group are also a great support for the past three years. Thanks to Gary Bromham, Bhusan Chettri, Emmanouil Chourdakis, Jiajie Dai, Beici Liang, Marco Martínez, Saumitra Mishra, Dave Moffat, Veronica Morfi, Inês Nolasco, Maria Panteli, Rod Selfridge, Dalia Senvaityte, Daniel Stoller, Dan Stowell, Mi Tian, Siying Wang, Changhong Wang, Will Wilkinson, Yongmeng Wu, Simin Yang, Luwei Yang, and Adrien Ycart. It has been a privilege to work along side of them. I enjoyed the working atmosphere and friendship during my time. I also received precious advices from the audio signal processing community and MIR community. The conversations and discussions happened during the conferences and workshops have intrigued many research ideas. Thanks to Dr. Justin Salamon who is my mentor on the WiMIR mentor program. Thanks to the fellow researchers in these areas for sharing knowledge and presenting ideas. It has been a tremendous help along my research training.

I would like to thank the professors, lecturers, and fellow students at both Electrical and Electronic Engineering Department at Imperial College London and the Information Science and Electronic Engineering School at Zhejiang University China. The education I received in both schools encouraged me to pursue further education in research. I was

more than lucky to study my Bachelor degree at Zhejiang University. It nurtures every students' dream and helps to make them possible. I would always be grateful for it.

I would like to thank my colleagues of my summer internships. I spent two terrific summers in BBC and Alibaba. Thanks to everyone in BBC Datalab for the guidance on how to fit research ideas into a real world production environment. Thanks for all the exciting events you have organised. Thanks to everyone in AliMusic, Alibaba for teaching me lots of the engineering details. I love the reading club we kept for exploring new ideas. Thanks to the friends I made along the way. It was a pleasant surprise to meet like-minded people and build bonds within such a short period of time.

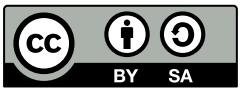
Thanks to my friends who shared my worries and happiness, especially my hiking group, game pals and lunch crew. You can always put a smile on me. Thanks to those friends who have time differences but never lost contact with me. Thanks to my anonymous friends on Douban for sharing news, views, and excellent debates. Thanks to my family for all the trust and support you have been provided. I want to thank my brother for being such a darling boy to share the burden which should have been mine.

Thanks to Edwin Tomboza, for the endless proofreading, for always being positive, for the unconditional emotion support, and many more. I can not express in words how important your support has been to me. This Thesis would not be possible without it.

Thanks to the music that kept me company during my PhD. They are the very reason that this research exists.

# Licence





# Contents

# List of Figures

# List of Tables

# List of abbreviations

| | |
|---|---|
| ADSR | Attack, Decay, Sustain, and Release |
| AFCC | Auditory Spectrum Cepstral Coefficients |
| DNN | Deep Neural Network |
| DAW | Digital Audio Workstation |
| DRC | Dynamic Range Compressor |
| EMD | Earth Mover's Distance |
| ERB | Equal Rectangular Bandwidth |
| EQ | Equaliser |
| FIR | Finite Impulse Response |
| GMM | Gaussian Mixture Model |
| HFC | High Frequency Content |
| IIR | Infinite Impulse Response |
| ISTA | Iterative Shrinkage Threshold Algorithm |
| KZ | Kolmogorov-Zurbenko (filter) |
| LTI | Linear Time-Invariant |
| MA | Moving Average |
| MAE | Mean Absolute Error |
| MDCT | Modified Discrete Cosine Transform |
| MSE | Mean Square Error |
| MFCC | Mel-frequency Cepstral Coefficients |
| NMF | Non-negative Matrix Factorisation |
| OOB | Out-Of-Bag (error) |
| PSOLA | Pitch Synchronous Overlap-Add |
| RIS | Random Iterative Search |
| RMS | Root Mean Square |
| SOLA | Synchronised Overlap-Add |
| STFT | Short-time Fourier transform |
| VST | Virtual Studio Technology |

# Chapter 1

# Introduction

Music is a widely loved art form with a long history. In the modern age, the storage, distribution, and consumption of music are mostly in digital format, not to speak of music mixing and production. Like many new expression forms emerged in the twentieth century, music mixing evolved along with the development of technology. It started from the invention of the multitrack tape machine, which makes recording different instruments separately possible. Audio effects are designed for individual audio track processing as well as multitrack mixing. With the digitalisation of the industry, especially the invention of Digital Audio Workstation (DAW) and standard audio processing plugin formats such as Virtual Studio Technology (VST), more advanced digital audio effects are developed as audio plugins for mixing and production. Early digital audio effects were inspired by hardware implementations of effects that used a physical process or analogue circuits to transform or enhance audio signals. Many digital signal processing techniques were developed for this purpose. With the increase in computational power, even physical modelling and circuit simulation became possible. Modern audio effects use a wide range of techniques to transform or enhance sounds. The processing power of computer gradually improved, and it leads to a further development of the digital audio effects. In the recent decade, thriving machine learning and artificial intelligence motivates a new research topic, *Intelligent Music Production*. This topic combines audio engineering, acoustic modelling, signal processing and machine learning together. High level control of audio effects becomes possible with the development of signal processing techniques and machine learning models.

This Thesis is under the scope of this topic. The author proposed an innovative intelligent system for controlling audio effects. The system is designed and tested primely on one

of the most complex and important audio effect, the Dynamic Range Compressor (DRC). This chapter provides the motivation and aim for this research. The contribution of this research has been summarised first, which is followed by a detailed Thesis structure. The publications that are associated to this research are provided at the end of this chapter.

## 1.1 Motivation

The production of music today is largely digitised, with the most commonly used tool for production and mixing is the DAW software. However music production still remains time consuming and labour intensive. Repetitive procedures still exist in this profession. This problem creates an opportunity to use intelligent tools to improve the process. A high level control mechanism is able to smooth the process for professional producers by releasing them from time consuming routine jobs. Similarly to other professions in the information era, intelligent tools can be built to assist music production. Such tools may also provide unexpected artistic inspirations.

Music production is a profession and craft that relies heavily on experience and know-how. It requires many years of training, while to become good at audio production or engineering requires both objective knowledge of the tools and the ability to make good aesthetic judgement. Therefore there is a high barrier between amateurs and professionals. The development of DAW and audio signal processing plugins bring the studio into bedrooms, which promotes many *bedroom producers*. Many famous artists, especially electronic and hip-hop musicians, started from making music using DAW and digital instruments [Vice, 2018]. This is an exciting trend in the music industry, especially since it encourages young people and hobbyists to create their own music with less effort. However the use of these convenient tools still requires deep understanding of the underlying signal processing. *Bedroom producers* still need to spend significant amount of their time on learning the underlying technologies rather than focus on their music. For casual users, the time and effort required might become a barrier they could not conquer, and the reason that stops them from making music. The author believes that further development of mixing tools should take this into consideration, and bring more intelligent tools for the casual users and hobbyists.

This Thesis is motivated by the possibility of improving the music production process. Intelligent tools can be beneficial for both professional and causal users. There are many

possible ways to improve this process. Before introducing the approach that is proposed in this Thesis, it is worth to mention how people normally use audio effects. Practitioners who use audio effects often follow an intuition. For instance, casual users of audio effects often lack practical experience or knowledge of low-level signal processing parameters of audio effects, therefore, they often describe a desired effect by providing an example, e.g., name a particular style, artist or song, instead of articulating specific features, e.g., "I'd like a short attack" [McGrath et al., 2016]. This Thesis proposes a control system that follows this intuition. The author proposes a new intelligent controlling mechanism. The system uses an audio example as reference. A series of experiments led to the development of an intelligent control mechanism for audio effects. An evaluation mechanism and a method for optimisation have also been developed for the proposed system. The details of the system design and experiments will be outlined in the following sections.

## 1.2 Aim

This Thesis has multiple aspects and focusses on several areas of related research. The first research target is at proposing an innovative system design. The proposed system in this Thesis is a new approach to interacting with audio effects. The author aims at bringing new ideas and new opportunities to the domain of Intelligent Music Production research. This Thesis is written to present the whole research cycle around the proposed approach, which includes system design, optimisation and evaluation. Since there is very little discussion on a similar approach in the literature, the author intends to provide a thorough discussion of the proposed solution. Discussion of the design process and the application scenario is provided. The author also aims to provide insights and analysis so that it would be easier for fellow researchers to adapt this approach to other audio effects or derive a more efficient system.

## 1.3 Contributions

This Thesis is aiming at contributing to the research area of Intelligent Music Production. It is a fairly new area but has a huge potential. More research and discussion are needed to push forward the development of this research area. This Thesis therefore aims to evoke discussion in a new mechanism of controlling audio effects.

First, the main contribution of the Thesis is the system design. It presents the de-

sign, analysis, evaluation, and optimisations of a system. Second, this Thesis contributes to audio feature engineering for audio effect parameter recognition as well. The relations between low level audio features and DRC are analysed. New audio features are introduced for specific DRC parameters. The author also trains a Deep Neural Network feature learning model to generate an universal and robust feature embedding. Finally, this Thesis contribute to the audio similarity research domain. There has been very little discuss in the literature on the audio dis-similarity caused by audio effects. Yet it is a valuable research topic as there are many potential applications. For example, a method for audio sample retrieval for professional music library that contains samples processed by different effects. This Thesis provides a model to measure the similarity caused by DRC. The Thesis also contributes to the idea of optimising the accuracy of the predicted audio effect parameter prediction using a multi-stage search algorithm.

This Thesis presents a listening test focused on the audibility threshold of the ballistic parameters of DRC. The listening test on audibility threshold is a highly complex test, therefore, substantial work is required. The experiment presents in this Thesis is served as a support of the results out of the computational model. It can also be considered as a starting point of a series of research in this direction.

## 1.4 Thesis structure

**Chapter 1 Introduction**

In this chapter the author presents the context and motivation of this research. The research aims, contributions, and thesis structure are also outlined.

**Chapter 2 Background**

This chapter provides essential background needed in developing this Thesis. The discussion starts from a brief introduction of music production. This is an essential process for making music in the modern era. Music production involves music editing, mixing and mastering. The processing chain relies heavily on audio effects. Therefore, an overview of widely used audio effects is provided along with their underlying signal processing and basic use cases.

Intelligent Music Production is still a relatively new research area. The author therefore presents a walkthrough of previous research. A large part of this research requires

audio feature extraction and a vast variety of audio signal processing algorithms. Related research has also been included. Furthermore, a brief introduction of auditory models and similarity is provided.

This research is aiming at making an intelligent system, therefore, machine learning related algorithms are frequently used, including feature engineering, feature selection, regression models, and Deep Neural Networks. A brief description of these methods is also presented in this chapter.

**Chapter 3 Intelligent System Design**

The proposed system designs are described in detail in this chapter. It provides the motivation and inspiration of the design, and the reason why this research focuses on the Dynamic Range Compressor. A thorough workflow of the proposed system is provided as well as the design path. A simple dataset of isolated notes is generated and used to evaluate the efficiency of the initial design.

**Chapter 4 Feature Design and Selection for Mono-instrument Loops**

The key to the performance of the proposed system is the efficiency of audio features. This chapter provides a full list of features that the author considers relevant to the problem. It includes both conventional audio features and handcrafted features, which the former are the ones widely used from the literature and the latter are the ones designed in this research by the author. This chapter focuses on more complex audio material than the previous one, mono-instrument loops. Therefore, audio decomposition algorithms are considered as part of the feature design. Since it is possible that handcrafted features contain noise and redundancy, a feature selection scheme has been proposed and evaluated in this chapter as well.

**Chapter 5 Siamese Model for Feature Learning**

This chapter proposes another approach for feature extraction. The methods proposed in the previous chapters extract one set of feature for each parameter. This chapter aims at designing a universal feature extractor using a feature learning neural network. This feature learner will empower the system to predict several parameters jointly rather than separately. More complex audio materials, polyphonic music, are tested in this chapter and convincing results are produced using this approach that outperform the previous approaches in most cases.

**Chapter 6 Audio Similarity Model focusing on the Perceptual Aspects of Sound Modified by Audio Effects**

The author proposes an audio similarity model as an objective evaluation method for the intelligent audio effect control system. This chapter proposed a novel similarity model incorporating the use of auditory filters and loudness model. A substantial evaluation experiments have been run to test the efficiency of the model design.

**Chapter 7 Optimisation and Subjective Evaluation of the Intelligent System**

Two final experiments are presented in this chapter. In the first experiment, an optimisation method has been proposed and evaluated. This Thesis has proposed a prediction model and an objective evaluation function. The author assumes that using the objective evaluation function as an optimisation function will improve the prediction performance. The second experiment is a subjective listening test. It is designed to discover the minimal audibility distance between parameters. This knowledge can be used to inform if the prediction error of the intelligent model is audible. The two parts of the research has been put together because the results from both experiments can serve as a perceptual support for the proposed method of intelligent audio effect control. The first experiment in this chapter is an optimisation of the prediction model with an emphasis on the perceptual aspects. The second experiment is a perceptual test to support the prediction model.

**Chapter 8 Conclusion**

The conclusion of the Thesis is presented in this chapter. The efficiency of the audio features is important for the proposed system. Both handcrafted features and the learnt feature embeddings are proved to be able to improve the performance for audio in different complexity level. The components, e.g. filters from the computational auditory models are also useful in terms of evaluating audio similarity focusing on audio effects. The ideas of the future research are also outlined.

## 1.5 Associated publications

Portions of the work detailed in this thesis have been presented in national and international scholarly publications, as follows:

- Chapter 3: was published as conference paper in DAFx, 2017.

Di Sheng and György Fazekas. *Automatic control of the dynamic range compressor using a regression model and a reference sound.* In Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17), 2017.

- Chapter 4: Section 4.3 and Section 4.5.1-4.5.2 was published as a conference paper in ICASSP, 2018.

  Di Sheng and György Fazekas. *Feature design using audio decomposition for intelligent control of the dynamic range compressor.* In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 621 - 625. IEEE, 2018.

- Chapter 4: Section 4.4 and Section 4.5.3-4.5.5 was published as a complete-manuscript peer-reviewed convention paper in AES Convention, 2018.

  Di Sheng and György Fazekas. *Feature selection for dynamic range compressor parameter estimation.* In Audio Engineering Society Convention 144. Audio Engineering Society, 2018.

- Chapter 5: was published as a conference paper in IJCNN, 2019.

  Di Sheng and György Fazekas. *A Feature Learning Siamese Model for Intelligent Control of the Dynamic Range Compressor.* In the International Joint Conference on Neural Networks (IJCNN), 2019.

- Chapter 6: is under review as a journal paper in the Journal of the Audio Engineering Society.

  Di Sheng and György Fazekas. *Estimating Similarity Between Sounds Processed Through Audio Effects.* Under review of the Journal of the Audio Engineering Society (JAES).

- Chapter 7: Section 7.1 is submitted as a journal paper in the Special Issue "Digital Audio Effects" of Applied Sciences.

# Chapter 2

# Background

This research is built with the support of the experiences accumulated by previous researchers and audio engineers. In this chapter, the background and related work is summarised. The discussion begins with an introduction to music production. Introducing this as the context of application will help with the understanding of the system design, and many more research choices the author has made along this research period. An introduction of audio effects with an in-depth presentation of dynamic range compressor is also provided in Section 2.1. It is followed by an overview of intelligent music production, editing and automatic mixing, which involves previous works related to DRC in Section 2.2. In Section 2.3, related music features are outlined as well as the signal processing algorithms that helps to build the audio features in this work. The author goes on to summarise the auditory model and audio similarity algorithms that are related to the objective evaluation model this research has developed. A large part of this research relies on machine learning models and techniques. The usage of machine learning in audio signal processing is discussed, especially in the intelligent production field in Section 2.5.

## 2.1 Music Production

Music production plays a very significant role in modern music. The choices made by producers, to some extent, decide the music mood, the emotion it would provoke and also the response it should elicit. The same song can be dramatically different if produced or mixed by different producers and engineers. The reader can find two versions of *"All*

*Apologies"* by Nirvana mixed by Scott Litt[1] and Steve Albini[2] and feel the differences. A common workflow for music production is illustrated in Figure 2.1 [Izhaki, 2013]. The *Songwriting* and *Arranging* stages involve artistic creation and decisions. The *Recording* and *Editing* stages mainly focus on capturing the best quality sound. A good recording would provide more creative opportunities to the mixing engineers because there will be no need to think about fixing the flaws. Having a reasonable recording, the *Mixing* and *Mastering* stages are the ones that balance the tracks and add desirable effects on each group of tracks. These are the stages where audio effects are widely applied. A more detailed introduction of these stages are provided in Section 2.1.1.

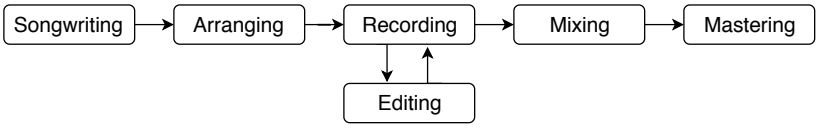



Figure 2.1: Common production chain for recorded music.

The main research subject in this Thesis is the area of audio effect. The implementation of audio effects involves a wide range of signal processing. A basic introduction of audio effects including classification and some of the underlying signal processing details is provided in Section 2.1.2. The detailed overview of Dynamic Range Compressor which is the main focus of this Thesis will be outlined in Section 2.1.3. The author also includes the basics of adaptive audio effects which inspires the design of the system proposed in this research.

### 2.1.1 Music Editing, Mixing, and Mastering

Music production is a complex process, and will be easier to understand if divided into multiple stages. Production normally happened after or along with the music recording session. There are many stages in Figure 2.1, and the ones related to this research are outlined in this section.

Editing is normally the final stage for audio engineers to prepare all the audio materials before sending them to a mix engineer. Music editing involves choosing the right recording takes as well as repairing any bad performance or recordings from the live recording stages. Mixing is a process that balances, treats and combines the multitrack materials into a stereo, sometimes mono or multi-channels signal. To start the mixing process, many audio

[1] `https://www.youtube.com/watch?v=1KPiElw6qy8`
[2] `https://www.youtube.com/watch?v=nu0rYx2wKX8`

engineers would start with a rough mix. A rough mix is an initial mix with little thought and less care than a final mix. It is a good starting point for audio engineers to become familiar with the structure of the material, the sound quality of the raw audio, as well as to capture the mood of the song. This is a stage where intelligent systems could be developed and where they fit the best in the music production workflow. There is no standard for "perfect" mixing. At mixing stage, the process needs to balance the loudness, avoiding masking effects across tracks, as well as to spread sound evenly on the frequency axis and sound field. In general, the fundamental problem for mixing is to shape the audio into a desirable sound. This is where normally audio engineers would use a reference to describe a desirable sound. The final stage of music production is mastering. This is a stage where the audio engineers create the final mix suitable for all kinds of playback systems. This stage requires choosing the right amount of equalisation and dynamic range compression to perfect the final version of the music. Most of the tasks in these stages can be achieved through audio effects.

### 2.1.2 Audio Effects

Due to technical and aesthetic requirements, the music industry developed an increasingly large number of tools for the manipulation of audio content to achieve desired sound qualities. Changing the dynamic range, timbre or frequency balance of recordings have first become widely possible with the introduction of analogue signal processing techniques, for instance, linear filters like the Equaliser and non linear process like the Compressor. Digital technologies such as software plug-ins or audio effects embedded in Digital Audio Workstations have significantly extended and, to some extent, replaced analogue effects. Digital audio effects play an essential role to shape a desirable sound in the post-processing of music nowadays. They have also evolved with the new technology, e.g. adaptive real-time DSP techniques have been included in the design of effects. Due to the plethora of audio effects, it is not easy to classify them into general categories. Some previously classification methods will be introduced in this section. In the view of signal processing, especially in this research, commonly used effects can be classified as time domain, frequency domain, and time-frequency domain effects [Zölzer et al., 2002]. They can also be classified as linear and non linear systems. Examples given in Table 2.1 and Table 2.2.

Users will have different demands for the audio effects with respect to their roles. Audio engineers or researchers are more likely to pay attention to the signal processing aspects,

| | |
|---|---|
| Time domain | Dynamics(Compressor; Extender; Limiter; Noise Gate;...) |
| | Delay-Line(Vibrato; Slapback; Echo;... Filter: Flanger, Phaser, Wah-wah... |
| | SOLA; PSOLA(Time/pitch shifting; Time scaling;...) |
| | Time-shuffle; Resampling; Amplitude modulation;... |
| Frequency domain | Equaliser; Spectrum non linear modification; Spectrum envelope modification; Morphing; Robotisation; Whisperization;... |
| Time-Frenquency domain | Phase vocoder; |

Table 2.1: Audio effects classification I, classifying the common effects according to their domain of application.

| | |
|---|---|
| LTI | Equalisation; Filter; Spectrum envelope modification; |
| Non Linear | Dynamics effects; Adaptive filter;... |

Table 2.2: Audio effects classification II, classifying the common effects by linear and non linear process. LTI stands for Linear Time-Invariant.

whereas, composers and performers will focus on perceptual aspects. From a different point of view, audio effects are more than the underlying signal processing but are also designed to manipulate audio perception. Thus, another way to classify them is based on perceptual attributes. There are several common categories at the perceptual level: loudness, time/rhythm, pitch, spatial hearing and timbre. Table 2.3 illustrates a rough classification.

| | |
|---|---|
| Loudness | Dynamics; Gain control; Phrasing(legato, pizzicato); tremolo;... |
| Time\Rhythm | duration, tempo, rhythm related effects: accelerando, deccelerando; time-scaling; |
| Pitch | Pitch shifting; Autotune; other effects related to chroma;... |
| Spatial Hearing | Doppler; Reverberation; Panning... |
| Timbre | Vibrato and other modulation related effects; Chorus and other delay line effects; Equaliser; Whisperisation; Adaptive filters; and much more... |

Table 2.3: Audio effects classification III, classifying the common audio effects by the perceptual attributes they can alter.

In most cases, one effect can be used for different purposes. For example, the Compressor is designed to manipulate dynamics, while it can also be used creatively which leads to some artistic effect. It can add a punch effect or create a more powerful and thicker sound

in pop music [Huber and Runstein, 2010]. A single system of categorisation is not sufficient as many effects affect multiple musical and perceptual dimensions. The introduction of interdisciplinary classification addresses this problem by connecting different disciplines together, from acoustics and electronic engineering to psychoacoustics and music cognition [Verfaille et al., 2006a]. In this structure, three discipline-specific classifications are presented: they are based on the *underlying techniques*, *control signals*, and *perceptual attributes*. The interdisciplinary classification will place links between layers, and one effect can therefore fall into multiple categories.

Based on the idea of interdisciplinary classification, Zölzer et al. [2002] provides examples of how to classify Distortion, Equaliser, and the Wah-wah Effect. In a similar fashion, and based on the author's own understanding, a Compressor can be classified as Figure 2.2.

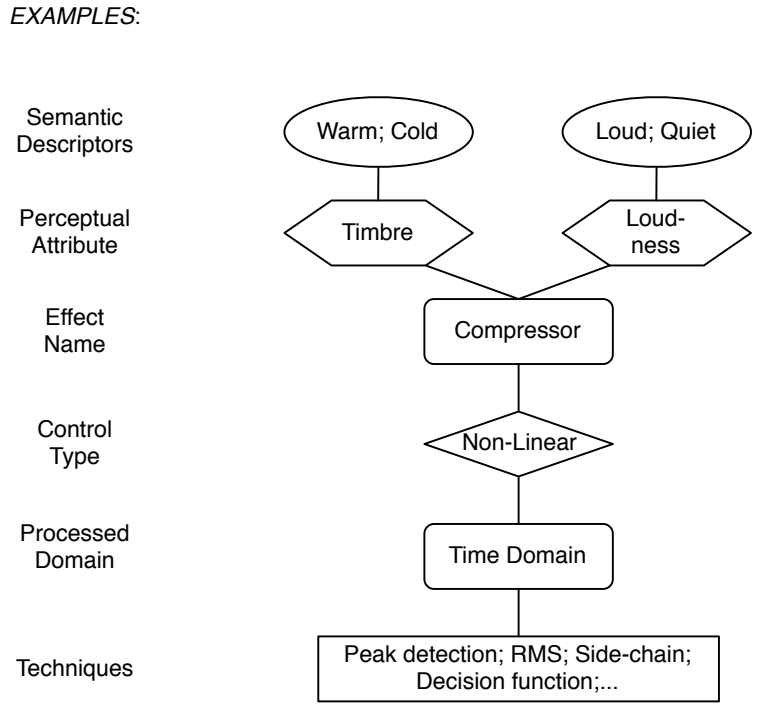



Figure 2.2: Interdisciplinary classification relations for Dynamic Range Compressor.

There are also *sound effects* that are mostly used in movies, television, and radio programmes. They are typically arranged in a library of recorded or synthesised audio. Cheer, applause, or laughter are the typical effects within this category [Cai et al., 2003]. Although they are very important effects and are frequently used in production, they are not necessarily related to music or altering existing sounds. Therefore, in this research they are

considered out of scope.

Manipulating a linear system, for example reverse engineering, is not a particularly hard problem. Therefore the intelligent control of these systems has already been studied relatively thoroughly. This PhD focuses on non linear audio effects. This research studies one particular non linear audio effect: the Dynamic Range Compressor. The study is also purely on the digital level of this effect and analogue hardware implementations are not discussed or analysed. The following subsections focuses on this effect and other higher control mechanism of audio effects.

### 2.1.3 Dynamic Range Compressor

DRC is a common tool of dynamics control for audio. It is wildly used in sound recording, music production, radio broadcasting, and live performance. The function of a simple compressor is to reduce the loud parts of the sound and amplify the quiet parts, which results in smaller dynamic range. This can be beneficial for the perceived loudness, or it can assist to reduce the coding rate for any transmission channel with a bandwidth limitation.

A compressor can be designed to compress loud sounds that pass a certain threshold; it can also be designed to amplify quiet sounds that lie below a certain threshold. The first type is called a downward compressor, and the latter is called an upward compressor. Here, the research will focus on downward compression but we assume that the same mechanics applies to upward compression.

Equation 2.1 describes compressor in general condition. $X$ represents the audio sample energy or the root-mean-square (RMS) energy, depending on the design. $G$, $G1$ and $G2$ are the gain factors of audio signal samples in $dB$. The threshold is one of the parameters that users need to configure. The signal above will be compressed according to the ratio, i.e. Equation 2.2. A more apparent illustration is provided in Figure 2.3. The threshold decides above which energy level the signal needs to be compressed. Ratio decides how much it will be compressed. For the sake of simplicity, the figure demonstrates the hard knee compressor. Soft knee will apply a smooth curve for the gain turning point, where the knee length will decide the curvature of the smooth curve.

$$G = \begin{cases} G1 & if \quad X < threshold \\ G2 * (X - L) & if \quad X \geqslant threshold \end{cases} \tag{2.1}$$

$$ratio = G2/G1, \quad normally \quad G1 = 1, \quad so \quad ratio = G2. \tag{2.2}$$

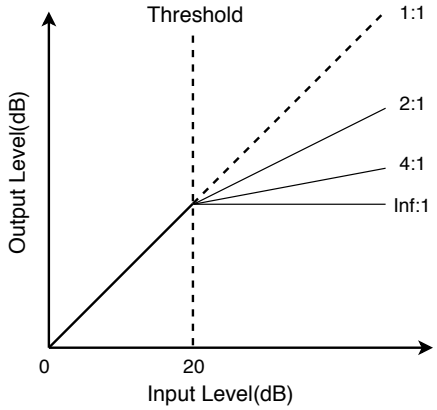



Figure 2.3: Dynamic Range Compressor gain control behaviour

where $G1, G2$ are signal gain, $L$ is threshold in dB scale, and $X$ is signal magnitude.

The attack time and release time are another two essential parameters of the compressor. They decide how fast the compression will operate or recover. If the compressor acts directly when the sound reaches the threshold, it can lead to some unnatural results that are not pleasant to listen to. The attack time is the approximate time for compressor to reach the threshold. The release time is the approximate time for the compression to recover. In implementation, these time constants are normally transferred through a smoothing factor as shown in Equation 2.3, where $\tau$ represents the attack or release time. Make-up gain is another commonly used parameter, which can be added to the output of the compressor to adjust the overall energy level. Figure 2.4 shows an example when applying DRC to a square wave. It illustrates the attack and release phase of the effect working along with threshold, ratio, and make-up gain.

$$\beta = 1 - \exp[-2.2/(f_s * \tau)] \tag{2.3}$$

Being familiar with the basic mechanics of the DRC is not exactly the same as knowing how to operate this audio effect. Similar to all audio effects, the DRC can be used in a technically correct way, as well as an artistic way. The latter may not be technically perfect, but can provide immense listening experience. The ordinary way is to use the compressor to control loudness. Loudness is a notion that people sometimes confuse with

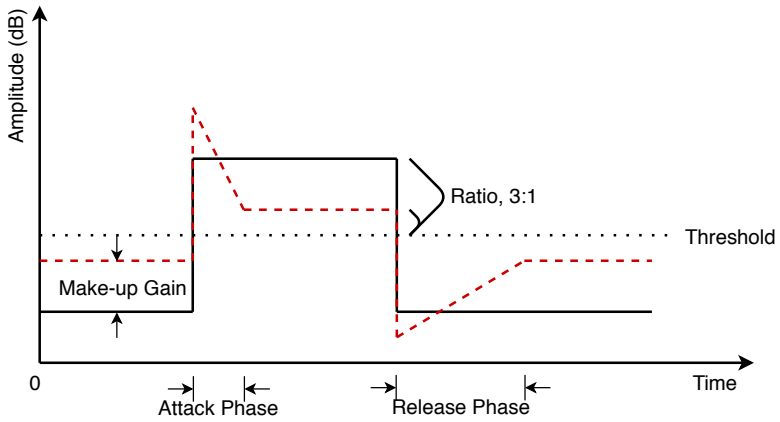



Figure 2.4: Illustration for some of the parameters of Compressor, when applying to a square wave.

the magnitude of the waveform. It is however a perceptual attribute that depends on the non linear auditory phenomena and therefore strongly influenced by the dynamic range. The control of loudness and dynamics can create a compact, thick, and powerful feeling, which is very popular in the music of the 21st century. A correct attack and release time for the compressor is also how the "punch" sound in pop music is created. Alternatively, one might use longer release time to compress one instrument track in order to smear the attack of another to create unexpected psychedelic perception or other artificial effects. In conclusion, the DRC is a very common audio effect, but the use of it can be various. The common use cases are listed in Table 2.4 [Izhaki, 2013].

| | |
|---|---|
| Accentuate the sound details | amplify the low volume, and reduce the large volume; |
| Balancing levels | multi-tracks balancing when one specific track is louder than others; |
| Increasing perceived loudness | smaller dynamic range sometimes means higher loudness perceptually; |
| Reshaping the dynamic envelope | changing attack or release time; possibly using to adjust tempo, add punch effect, or enhance decay, etc.; |
| De-essing | reduce the sudden "hissing" sound; |
| Ducker | side-chain compressor can control the loudness of one track based on the other track; |

Table 2.4: Compressor use case examples

DRC can be used as a single effect, as a parallel DRC or as a serial one. Parallel compression provides a louder sound while retaining their dynamics. Serial compressors

are able to separate multiple compression tasks into single compression tasks. It may also be used on a certain frequency band. These applications are out of the scope of this research.

### 2.1.4 Adaptive Audio Effects

Comparing with the usual audio effects, adaptive audio effects are able to provide a high-level control, or to create completely new effects. They are normally designed through combining an effect with an adaptive control method, more specifically, a time-varying control derived from audio features. Figure 2.5 shows a general structure of adaptive effect [Verfaille et al., 2006b, Zölzer et al., 2002].

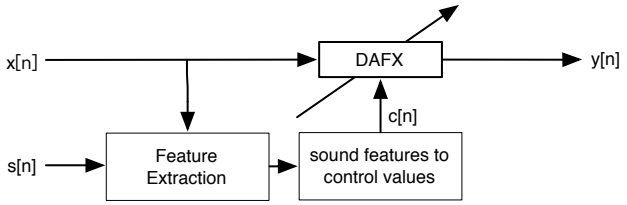



Figure 2.5: Adaptive audio effect structure

An adaptive effect can be controlled by the feature extracted from input signal $x[n]$, i.e. auto-adaptive. It can be controlled from another one or more input signal $s[n]$, for instance, in the case of a ducker (see Table 2.4), or features extracted from the input signal $s[n]$. This type of control scheme is referred to as external adaptive. The ones controlled by the output $y[n]$ are called feedback adaptive, and those which are controlled by both internal and external signals are cross adaptive effects. Adaptive effects can be extended to gestural control as well.

Simple examples in this adaptive effect category are auto-tune, compressors, cross-synthesis, and so on. There are also methods for commonly used effects, for example, adaptive Equaliser (EQ), adaptive panning, etc.

## 2.2 Review and Applications of Intelligent Music Production

Audio engineers play an important role in the music creation process. They are required in recording sessions, the mixing process, the mastering process, and many more. Audio

engineers have to shape the music track into a desirable sound, which involves, but not limited to, tune the single tracks to a pleasant timbre, balance the loudness level, frequency spread, and sound source positions of multi-tracks. With the development of digital signal processing techniques, this profession is being digitalised rapidly. However, some parts of the job still remains labour intensive and time consuming, for example, matching loudness level. These parts of the process do not require artistic creativity, which give researchers an opportunity to develop intelligent systems to assist the process. The essential tools for audio engineers in the digital era are Digital Audio Workstation (DAW) and digital audio effects. In the following sections, the previous applications of intelligent music editing, mixing and production systems with an emphasis on Dynamic Range Compressor are outlined.

### 2.2.1 Intelligent Editing Applications

In this section, the Intelligent Editing refers to the research area that designs and analyses tools, applications or functions that helps with the music editing. More specifically, "Editing" refers to the applications that present higher level audio features in music editing applications, i.e. adding semantic descriptors or simplified graphical interfaces. Loviscach [2008] describes a control method for Equaliser that allows users to draw points or use free-hand curves instead of setting parameters. Subsequent work [Kolhoff et al., 2006] provides a creative method to map features to shapes. However, the shapes of this system do not link directly with the "meaning" of settings, rather they are classifications of a group of presets. Cartwright and Pardo [2013] outlines a control strategy using descriptive terms such as "warm" or "muddy", and demonstrates a method of applying high-level semantic controls to audio effects. Wilmering et al. [2012] describes a semantic audio compressor that learns to associate control parameters with musical features such as the occurrence of chord patterns. This system provides intelligent recall functionality. Another type of the audio effects is identified as *metering and diagnostics*. It represents the applications which preserve the function of the audio effects but provide more information when analysing the high-level audio features on top of the traditional controls, e.g. [De Man et al., 2017] discuss an application that generates alerts when reverb reaches a certain level. Ford et al. [2015] provides a tool for analysing masking effects for instrument arrangement. The research mentioned in this section inspires design ideas by providing the solutions of using additional control for music signal processing.

### 2.2.2 Intelligent Production and Mixing

As discussed previously, automatic mixing and production applications can be divided into two categories. The first one can be described as a *blackbox* type of application, which refers to the case where applications leave no or little control to the users. Ma et al. [2015] provides an automation system that aims at finding the optimal dynamic range for each track considering domain knowledge gathered from audio engineers. The second category represents the ones that leave fewer controls to users compared to the original effect design, e.g. [Dannenberg, 2007, Liu et al., 2010, Giannoulis et al., 2013], which can be referred to as an *assistant* type.

Most of the control tools for DRC can be fitted into these two categories. The earlier applications in this field are mostly *assistant* type, which are designed to provide simplified or alternative control mechanism. Maddams et al. [2012] proposes a method to match loudness and loudness range across the multi-tracks leaving two control knobs for the user: automation level, and compression mode. Another approach proposed in [Giannoulis et al., 2013] requires threshold as input, and gives limited automation for the time constants and ratio. An alternative implementation of DRC using non-negative matrix factorisation (NMF) is proposed in [Sarver and Klapuri, 2011]. Here, the authors consider raising the NMF activation matrix to $\frac{1}{R}$ to obtain a compressed signal with ratio R after re-synthesis. This is viewed as a compressor without a threshold parameter. The more recent works are mainly *blackbox* applications. Authors propose a statistical method for intelligent compression in [Hilsamer and Herzog, 2014]. This method aims at matching the statistical moments with a reference audio, while missing the automation of the attack and release time. Mason et al. [2015] presents a personalised compression method controlled by the environmental noise. The method modified threshold and ratio using the loudness of noise while leaving the rest of the parameters fixed. However, this method has not been evaluated by subjective or objective test. A recent application applied deep neural network (DNN) in [Mimilakis et al., 2016]. It directly transfers the signal by using neural networks instead of predicting parameters and applies the learnt audio effect accordingly. An end-to-end approach is proposed in [Martínez and Reiss, 2018] where the method uses a generative model to simulate a certain configured audio effect. The results are promising, but as with other approaches using Deep Learning, the control over the model is very limited, and it appears to move the control problems from audio effects to

a neural network. From the feature learning perspective, some works learn the relevant features solely based on the audio to be compressed [Maddams et al., Giannoulis et al., 2013, Mason et al., 2015]. While Mimilakis et al. [2016] applies machine learning methods using compressed and uncompressed audio as training set. This work aims at learning the parameter configuration, or "artistic choices" from the data sets generated by experts. The system is fundamentally a rule based system but instead of having rules defined by human, the rules are learnt by a DNN. The problem with this approach is that the training set normally requires human effort to build. Moreover it is usually small and not easy to obtain. Hilsamer and Herzog [2014] has the most similar approach to the method proposed in this Thesis, since they apply a statistical model to match the statistical characteristics of one audio to the reference, while this Thesis uses machine learning model trained on audio features to predict DRC parameters' configuration in order to make one audio sound as close as possible to the reference.

## 2.3 Audio Features and Audio Signal Processing

The Thesis proposes a solution to a problem that fundamentally falls into the domain of audio signal processing. As many research in this area nowadays, this work takes the approach that involves the design and extraction of efficient features, and then uses them to train machine learning models to solve research tasks. Key to the performance in this approach is the effectiveness of the features and their relevance to the task. In this section, an overview of audio features and audio signal processing will be given first in Section 2.3.1. This is followed by the details of audio features related to DRC in Section 2.3.2. There are handcrafted features in this research specifically designed to represent compression related audio characteristics, and their design require the assistance of audio decomposition algorithms in cases where audio events overlap. Details of three types of decomposition algorithms are provided in Section 2.3.3.

### 2.3.1 Overview

There is a great variety of audio effects. Different audio features are designed to suit various purposes. In general audio features can be classified as low level and high level features. The former are normally directly computable signal-level attributes, i.e. temporal features and frequency features. The latter focuses on the perceptual level, i.e. those that require

perceptual or machine learning models to characterise perceptual musical or semantic descriptors. To provide a few examples, temporal features include, for instance, statistical features, i.e. mean, variance; zero-crossing rate; and energy features, e.g. Root mean square (RMS) curve. Frequency domain features include the signal bandwidth, spectral centroid, etc. An important feature set which is commonly used in prior works related to speech signal processing is Mel-frequency Cepstral coefficients (MFCC). This parametric description of the spectral envelope has the advantage of being level-independent and of yielding low mutual correlations between elements in an MFCC vector for both speech and music [Logan et al., 2000, Breebaart and McKinney, 2004]. Harmonic features consist of chroma-based features. The chromagram is derived from the spectrogram, such that it aggregates all spectral information that relates to a pitch class or tone height into one single coefficient [Müller, 2015]. The author will not discuss these types of features in detail in the following subsection since harmonic features are not very relevant to DRC. Perceptual features, or others may be referred to as psychoacoustic features, examples include description terms like roughness, loudness, or sharpness. Normally they can not be described by a single low level feature, but require a model for each corresponding descriptor, e.g. roughness model by Daniel and Weber [1997] and loudness model by Moore [2014]. More details about the perceptual features that are related to DRC are given in Section 2.4.1.

Most music related research tasks rely on signal processing algorithms. From using Fourier transform and its derivatives to extract frequency domain information, to music structure retrieval, e.g. chord recognition, onset event detection, rhythm detection, and source separation. It is not feasible to cover all algorithms in this field in this chapter. In the following sections, the author will focus on the audio features and signal processing algorithms that will be used in this research. The algorithms and equations are presented in detail in the following chapter where they are applied.

### 2.3.2 Audio Features for Dynamic Range Compressor

DRC is an effect to alter the dynamic range of audio. Changes in dynamic range can impact certain statistical characteristics of the signal, therefore these can represent changes induced by the DRC to an extent. Statistical moments can be used on the audio sample level or across larger blocks using frame-wise audio signals to devise features related to dynamics. Starting with the frequency domain, spectral centroid is a commonly used statistical measure computed from the spectra. It indicates the “center of the spectrum”

and can be used to describe the "brightness" of the sound [Schubert et al., 2004]. Centroid is the weighted mean spectra, i.e.

$$\frac{\sum_{k=0}^{K-1} f_k * Y(k)}{\sum_{k=0}^{K-1} Y(k)}, \tag{2.4}$$

where $f_k$ is the center frequency of a Short-time Fourier transform (STFT) bin, and $Y(k)$ is the magnitude of bin $k$. Higher order moments such as variance, skewness and kurtosis can also be defined to characterise spectra. Another popular frequency domain representation is the Mel-spectrogram. It is calculated by passing spectrogram through Mel-scaled filter banks. Mel-scale is a perceptual unit or a scale of pitches judged by listeners to be equal in distance from one another. It is a non linear scale devised to simulate the non linear perception of pitch in the human auditory system. It is most famously used to calculate Mel-frequency cepstral coefficients (MFCC). In the implementation of MFCC, frequency is converted to Mel scale using the following equations: $M(f) = 2595 \log_{10}(1 + f/700)$ or $M(f) = 1127 \ln(1 + f/700)$ [Ganchev et al.]. Triangle filter banks with center frequencies arranged according to Mel bands are applied to the spectrogram. The resulting coefficients are logarithmically compressed and this is followed by the Discrete Cosine Transform (DCT). In most prior works in speech recognition and music information retrieval, the top 13 coefficients are used typically as MFCC vectors. Fewer or a larger number of coefficients are used in some cases, balancing the size of the feature space with the level of detail in the spectral envelope representation. If the DCT and log compression are not applied to concentrate the energy, it is called Mel-warped spectra or simply a Mel-scale spectrogram. Higher order statistical moments can be used in conjunction with Mel-scaled spectra to calculate dynamic range related features. In this way, both perceptual information and frequency domain information are included.

Temporal features are equally important in this research. They are either based on audio samples, or energy envelope e.g. calculated using Root Mean Square (RMS) energy on a short time frame basis. Statistical moments can be applied to audio samples directly or a series of RMS energy measurements, where the RMS is calculated using Equation 2.5. $N$ is the frame size and $x_n$ is the audio sample amplitude in this equation.

$$X_{RMS} = (\frac{1}{N} \sum_{n=0}^{N-1} x_n^2)^{\frac{1}{2}} \tag{2.5}$$

Statistical calculation can be applied on time domain signal as well. For example, the

mean, variance, and higher statistical moments can be extracted from short time framed audio samples or RMS signals. They are also directly related to the dynamics of the signal.

Another common feature often computed in the time domain is the well-known ADSR (Attack, Decay, Sustain, Release) envelope structure. Since the DRC has parameters, often referred to as ballistics, alerting the attack and release phase of the audio, the author designed specific features based on the note structure. Details of this will be discussed in Chapter 4. A diagram of the ADSR is provided in Figure 2.6.

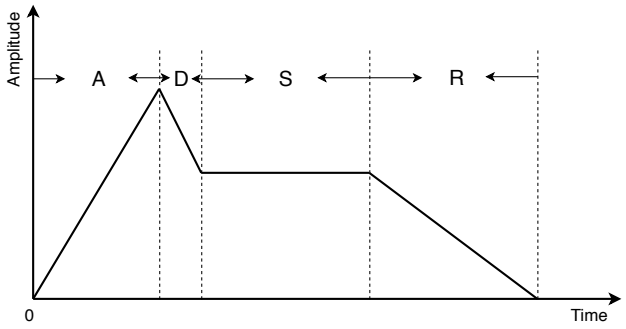



Figure 2.6: ADSR (Attack, Decay, Sustain, Release) envelope structure

Crest factor is a low level audio feature that is strongly correlated to dynamics [Peeters, 2004]. Crest factor is calculated by dividing the peak amplitude of a signal by the RMS value of the signal. It can be applied to the whole signal to measure the global dynamics or applied to windowed signal to represent the local dynamics. Equation 2.6 represents the formula of this feature, where $X_{RMS}$ can be referred to Equation 2.5. This feature is commonly used for the analysis of the DRC [Schneider and Hanson, 1991, Ma et al., 2015, Giannoulis et al., 2012].

$$C = \frac{|X_{peak}|}{X_{RMS}} = \frac{\max_{n \in N} |X(n)|}{X_{RMS}} \tag{2.6}$$

### 2.3.3 Audio Decomposition

In this research, the author applies the conventional audio features along with the features designed specifically for representing changes in the parameters of the DRC. Handcrafted features are strongly dependent on the note envelope structure, c.f. Chapter 4. With the increase of the complexity of the audio, the note structures become less easy to capture. There are different solutions for different levels of complexity. For mono-instrument loops, the author considers to apply audio decomposition before extracting handcrafted features.

There are three algorithms within consideration. The most straightforward method is based on onset event detection. Separate loops by detected onsets can help with extracting note structures. Guidelines for choosing the appropriate detection function is provided in [Bello et al., 2005]. Time domain methods are normally adequate for percussive signals, while spectral methods based on spectral or phase difference are suitable for pitched instruments. Complex-domain spectral difference [Bello et al., 2004] works well but with higher computational cost while state-of-the-art methods using deep learning [Schluter and Bock, 2014] performs better for polyphonic material. Since mono-timbral loops do not have such complex structure, the author opts for the simple High Frequency Content (HFC) [Bello et al., 2005] detection function as a starting point. The HFC function is constructed by summing the linearly-weighted values of the spectral magnitudes, such as: $D_H[n] = \sum_{k=0}^{N} k|X_k[n]|$, where $X[n]$ is the STFT of a time domain signal $x$. $X_k[n]$ is the value of the $k^{th}$ bin of $X[n]$. A temporal peak picking process is applied thereafter to detect the onsets.

The second approach is based on source separation using Non-negative Matrix Factorisation (NMF) [Berry et al., 2007]. NMF is a type of algorithm that factorises a matrix $V$ into two matrices $W$ and $H$, with the property that all three matrices have no negative elements. There are many approaches to achieve this. The author's research follows a popular and simple implementation, i.e. using a multiplicative update rule proposed by Lee and Seung [2001] to iteratively decompose a magnitude spectrogram into audio event templates and activations. NMF has shown to be a powerful decomposition for multivariate data [Lee and Seung, 1999]. It has an increasing use in the audio domain. It can be used on multiple levels. Wang et al. [2017] uses NMF on mono-instrument loops to decompose them into notes. Bertin et al. [2007] also provides an NMF based algorithm to decompose complex audio into activation patterns.

The third approach the author considered is transient/stationary audio separation. Researchers used orthogonal wavelet bases to separate transient in [Evangelista, 1993], while others combined Modified Discrete Cosine Transform (MDCT) and wavelet bases [Daudet and Torrésani, 2002]. A more recent work using the Iterative Shrinkage Threshold Algorithm (ISTA) [Siedenburg and Doclo, 2017] shows state-of-the-art performance, therefore, this method is selected as an additional audio decomposition approach. ISTA was first proposed for linear inverse problems in signal processing [Boyd et al., 2011]. It aims at reversing a process applied to a signal. Similarly to NMF, by recovering the processed signal, the processing matrix can be obtained, and therefore, can be used to compute audio

transient/stationary separation.

Most of the decomposition algorithms work better when the signals are smooth. The commonly, or traditionally, used smoothing filters are FIR/IIR filters, e.g. moving average (MA) filters. A MA filter is normally a simple lowpass filter. It takes $L$ samples of input at a time and takes the average of $L$-samples and produces a single output. The Kolmogorov-Zurbenko (KZ) filter [Yang and Zurbenko, 2010] that is applied in this research (see Chapter 4) is a series of iterations of a moving average filter of length $M$. A MA filter can be described as Equation 2.7. The KZ filter is described in Equation 2.8. As it can be implied from the equation, $M$ needs to be a positive, odd integer.

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} x[n-k] \tag{2.7}$$

$$y[n] = \sum_{s=-k(m-1)/2}^{k(m-1)/2} x[t+s] \times a_s^{m,k} \tag{2.8}$$

where $a_s^{m,k} = \frac{c_s^{m,k}}{m^k}$ and $c_s^{m,k}$ are the polynomial coefficients which can be obtained from equation:

$$\sum_{r=0}^{k(m-1)} z^r c_{r-k(m-1)/2}^{k,m} = (1 + z + ... + z^{m-1})^k \tag{2.9}$$

Equation 2.8 resembles to Equation 2.7. The difference is that KZ filters have a weight multiplied with each sample within the filter length. It can be seen as a repetitive MA filter, as the equation can be rewritten as an iteration of MA filtering. It has a better performance in terms of attenuating the frequency components above the cutoff frequency.

## 2.4 Auditory Models and Audio Similarity

To recap the discussion from Chapter 1, this research proposes a similarity model as an optimisation as well as objective evaluation tool for the intelligent control system. The model will be used to test if the output audio is similar to the reference compared to the original. This section provides the essential background for the similarity model.

Audio similarity is a broad subject because audio, especially music, is more than just signal, but it introduces emotion, related to culture, and has perceptual qualities. Audio similarity discussed in this section is focused on capturing the change in perceptual at-

tributes of sound due to the application of DRC. Perceptual attributes are not simple to capture or define. To capture them, computation models that simulate the human auditory system are needed, i.e. auditory models. In the following order, Section 2.4.1 outlines the auditory filters and loudness model that are applied in this research. Section 2.4.2 discusses some related work in audio similarity.

### 2.4.1 Auditory Filters and Loudness Model

As discussed in the previous subsection, the audio similarity, more specifically, dis-similarity, in this research relates to the subtle perceptual differences caused by an audio effect. To have a better representation of the perceptual attributes, it is reasonable to consider auditory models. Before discussing auditory models, the author needs to clarify the perceptual attributes this research focuses on. There are a vast variety of perceptual attributes and they are not all necessarily altered by audio effects. There is also no universal auditory model to represent all perceptual attributes yet. Audio effects typically alter several perceptual attributes including loudness, pitch, time, spatial hearing, and timbre [Amatriain et al., 2003]. In [Wilmering et al., 2013], the authors conduct a listening test to demonstrate that most of the audio effects are able to modify at least two aspects. In the case of the DRC, the primary attribute is loudness while the secondary attribute is timbre. This assumption is supported by the listening test. There are also logical arguments for explaining certain behaviour of audio effects. Reducing the dynamic range reduces the crest factor and allows for increasing the average gain. This results in an increase in perceived loudness [Wendl and Lee, 2014]. The influence on timbre is more subtle. Due to non linear signal processing, compressors *colour* the sound, while different designs even have a *signature* sound [Moore et al., 2016] that is typically recognised by engineers with critical listening skills. Different parameter settings of the DRC can also lead to timbre changes. Setting the attack and release times appropriately is how the "punch" sound in pop music is created for example, while using a longer release time to compress one instrument track in order to smear the attack of another can create unusual perception or other artificial effects [Izhaki, 2013].

To model music perception, researchers have developed many computational models to simulate the behaviour of the human auditory system. A brief introduction to the human auditory system is provided. The human ear has several components: The outer ear is the sound collector with the help of the pinna. The middle ear contains the ear canal and

the ear drum. They are used to pass and amplify sound and convert sound to mechanical vibration. The inner ear contains the cochlea and other components. They pass and filter the vibration and convert them into neural signals. A diagram of the auditory system is illustrated in Figure 2.7. In this research, we do not consider and discuss the neural excitation patterns in the central auditory system. Apart from the structure, there are also auditory phenomenas, e.g. critical bands within the cochlea [Fletcher, 1940] and non linear behaviour in pressure sensitivity leading to perceived loudness [Moore, 2014]. Most of the processing within the ear is non linear, for example, the critical bands are distributed on a non linear frequency scale as well. Based on these research, there are computational models designed to simulate the response of the human auditory system.

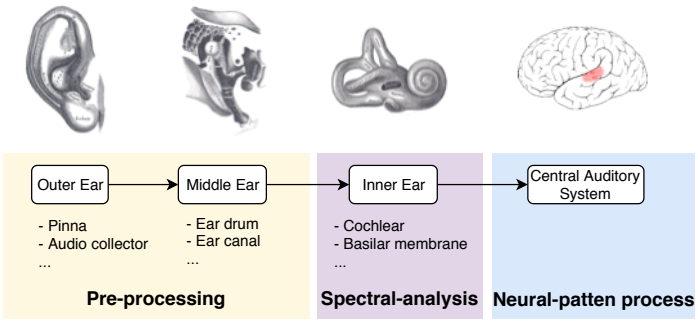



Figure 2.7: A brief diagram of human auditory system

As this research focuses on DRC, only the perceptual attributes related to DRC are considered. As it is explained at the beginning of this section, the two main perceptual attributes related to DRC are loudness and timbre. Because these are both perceptual qualities that depend on the behaviour of the auditory system, they are not directly measurable from the waveform or audio signal without the use of perceptual models. These models are designed to simulate the structure of ear, the transmission of hearing nerves, as well as the triggering of auditory neurons. Hearing functions are usually replicated by auditory filters. The construction of such filters have been the subject of studies over decades to construct models that fit psychophysical models of the cochlea as well as human listening experiments.

The Gammatone family of filters introduced in [Patterson et al., 1987] have become widely used to model the cochlea response. Equation 2.10 defines the impulse response of Gammatone filters, where $b = 1.019 * ERB(f) = 1.019 * (24.7 + 0.108f)$ [Slaney, 1993]. To improve simulation and provide a better fit to psychophysical data compared to previous approaches [Unoki et al., 2006], the Gammachirp filter was introduced by Irino and Patter-

son [1997] as a theoretically optimal auditory filter derived from the Gabor function, which is known to be able to achieve minimum uncertainty in a joint time-scale representation [Gabor, 1946]. The Gammachirp filters also have a sharp drop in the high frequency part compared with Gammatone filters, which are able to reflect the temporal masking effects of the auditory system better. The Gammachirp filter is described in Equation 2.11, where $\sigma$ is a time constant and $\phi$ is the phase. In both equations, $f$ represents the center frequencies placed along the Equal Rectangular Bandwidth (ERB) scale. Similarly to the Mel-scale introduced in Section 2.3.2, the ERB scale approximates auditory bands [Moore and Glasberg, 1983]

$$g(f,t) = t^3 exp(-2\pi bt) cos(2\pi ft) \tag{2.10}$$

$$g(f,t) = exp(-t/2\sigma) cos(2\pi(ft + c/2t^2) + \phi) \tag{2.11}$$

There are several works simulating the middle ear filters as well, c.f. Allamanche et al. [2001]. This filter has been suggested to apply in many research. For instance, Pampalk et. al. considered outer to middle ear transmission as well as the frequency masking effect in their work regarding audio similarity [Pampalk et al., 2008]. The filter response is defined in Equation 2.12.

$$H_{dB}(f_{kHz}) = -3.64 \times f^{-0.8} + 6.5 \times exp[-0.6 \times (f - 3.3)^2] - 10^{-3} \times f^4 \tag{2.12}$$

Another notable area of research is auditory modelling focusses on the perception of loudness. In terms of estimating perceived loudness, Moore and his colleagues have been developing a series of loudness models since the 90's [Moore and Glasberg, 1996], [Moore and Glasberg, 1997], [Moore and Glasberg, 2007]. A thorough review and a walk-through the development over several decades is given in [Moore, 2014], starting from models for stationary sounds to time-varying sounds, as well as models for impaired hearing. As introduced in Chapter 1, there is an audio similarity model designed in this Thesis. Both auditory filters and loudness models are considered to serve as the signal preprocessing step in the proposed similarity model. More details can be found in Chapter 6.

### 2.4.2 Audio Similarity

Audio and music similarity is a broad area of research with many existing models and algorithms concerning problems in difference levels and modalities. The authors in [Kaminskas and Ricci, 2012] provide a thorough review in the context of content-based music information retrieval. Although most algorithms focus on polyphonic music, they still provide useful insight of the problems addressed in this Thesis. In content based audio similarity, most systems rely on statistical models fitted on low or mid-level audio features. In the context of music tracks, Gaussian models trained on cepstral coefficients have been shown to work well in [Aucouturier and Pachet, 2002]. This has been considered as baseline in several subsequent studies. A similar approach was introduced earlier in [Logan and Salomon, 2001] with a histogram-based probability distribution instead of a Gaussian model. A substantial part of this Thesis is concerned with mono-instrument audio materials. There are related works for the similarity of individual instrument samples. Researchers outline a model for drum samples in [Pampalk et al., 2008]. Even though music similarity is a widely discussed subject, similarity for particular production features has not been explored. The differences caused by audio effects like the DRC are relatively subtle. Audio effects are mostly designed to alter the perceptual attributes, c.f. Section 2.1.2. Therefore, it is important to consider perceptual features when designing the similarity models targeting audio effects. There are works taking audio perception into account, for instance, Terrell et al. [2012] investigates the dependency of similarity models on listening conditions. The authors proposed level dependent auditory spectrum cepstral coefficients (ASCCs) to improve on MFCCs by making the system more perceptually relevant. Other similarity methods that consider symbolic information such as lyrics or metadata are out of the scope of this Thesis because this research only focuses on audio signal level similarity.

Even in audio signal level, there are many similarity algorithms targeting different problems. For example, genre classification, instrument recognition, and music fingerprinting, etc. Since our problem is very specific, there are not much previous research focused on the similarity caused by the application of audio effects. However, there are also works in the related areas we can gain useful insights from. Some research related to audio timbre similarity is reviewed next. Tzanetakis and Cook [2002] provides an automatic genre classification method. It averages the timbre features of a certain genre, therefore, each music piece can be matched to a group of music pieces. A critical review of genre clas-

sification can be found in [Sturm, 2012a]. Another related area is instrument recognition aiming at grouping similarly sounding instrument samples based on timbre. A summary of instrument recognition systems can be found in [Fu et al., 2011], and the commonly used instrument classification techniques are outlined in [Herrera et al., 2000]. Some similarity research focuses on relative similarity rather than absolute similarity. They may need to compare one music piece to another. Welsh et al. [1999] provides a timbre similarity algorithm, which extracts 1248 timbre features per song, and uses k-nearest neighbour to retrieve similarity songs. Timbre related features are frequently used in related research as well as statistical models to capture global characteristics of the audio that are important in the estimation of similarity. Some models important in this context are reviewed next.

Audio features are commonly extracted from a short time window of the audio signal. This is often based on the assumption that the signal, or some important aspect of it, is stationary for a short period of time. Additionally, when designing features, it is often the case that a local measure or estimate of some audio characteristic is sought for. However, these short-time features are unable to represent long term structural or global characteristics of the audio. To solve this problem, statistical aggregates such as mean and variance may be used if the distribution of a feature is known. For complex or unknown feature distributions, it is necessary to develop more complex statistical models, with parameters learnt or estimated from data. Statistical models represent probability and the divergence between probability distribution may be used as a proxy for estimating the similarity between those distributions, hence they are a proxy for audio similarity. Logan and Salomon [2001] proposed an approach, which is based on Gaussian Mixture Model (GMM) of MFCC features. GMM is a commonly used probability model to represent an audio signal. As audio signals are sometimes more complex than a single Gaussian distribution, multiple Gaussian can be a better representation. A possible similarity measure method between Gaussian models is called Earth Mover's Distance (EMD) which is provided by [Rubner et al., 2000].

Kullback-Leibler divergence (KL divergence) is a popular way to measure how one probability distribution is different from another. More specific, for discrete probability distributions $P$ and $Q$ defined on the same probability space, KL divergence can be calculated as Equation 2.13. KL divergence is often applied to compute the divergence between Gaussian distributions for the measurement of the information loss when using one distribution to approximate the other. Compared to estimating similarity using other distance

metrics, the success of using the KL divergence is crucially linked to it asymptotically going towards infinity when one of the distribution goes towards zero [Jensen et al., 2009]. Alternative methods include the Jensen-Shannon divergence [Fuglede and Topsoe, 2004], which is a symmetric version of the KL divergence. The divergence between mixtures of Gaussian models is not analytically tractable, therefore the similarity estimation method proposed in this Thesis uses the approach proposed in [Hershey and Olsen, 2007] which is based on variational Bayes approximation.

$$D_{\mathrm{KL}}\left(P||Q\right) = -\sum_{x \in X} P(x) \log(\frac{Q(x)}{P(x)}) \tag{2.13}$$

Such algorithm performs well in modelling timbre similarity focussing on the spectral envelope [Aucouturier and Pachet, 2002] although with some limitations, for instance, similarity estimation using this method may be perturbed by modification of the spectral envelope using equalisation in the audio domain [Sturm, 2012b]. However, this may be detrimental if the objective is to estimate a high-level semantic descriptor such as genre, but beneficial for measuring the impact of audio effects, since similar methods are sensitive to these types of transformations.

## 2.5 Machine Learning

A large part of this Thesis relies on machine learning techniques. It is a popular trend in audio signal research that low level audio features are engineered to complement machine learning models designed to perform well in high level tasks. Many parts of this research benefit from machine learning techniques in different ways. Firstly, a predictor that learns the association between audio features and DRC control parameters and provides for the estimation of optimal parameters are required in this research. This work uses a trained regression model as the predictor. The details of training the model and using it for DRC parameter estimation are presented in Chapter 3. Feature selection as a commonly used machine learning technique for data preprocessing will be beneficial to form an efficient feature set, since the audio features as outlined in Section 2.4 have a great variety and may overlap in terms of the audio characteristics they describe. Related work is provided in Section 2.5.2. Deep Neural Networks (DNNs) have become exceptionally successful for many research areas including audio signal processing. This research applied a DNN feature learning model as an improvement on the handcrafted features. The essential background

of the models this research applied is presented in Section 2.5.3 and the proposed method is discussed in Chapter 5.

### 2.5.1 Regression

Regression is a form of supervised machine learning. The goal of regression is to predict the value of one or more continuous target variables $t$, given the value of a D-dimensional vector $x$ of input variables. A commonly used model is linear regression. It aims to fit a linear function to represent the relation between the targets and inputs variables. More generally, a regressor aims to model the predictive distribution $p(t|x)$ as the uncertainty about the value of $t$ for each value of $x$. From this conditional distribution users can make predictions of $t$ for any new value of $x$. Regression models are trained in such a way that minimises the expected value of a suitable loss function. Normally the loss function for regression is the squared loss. To be concise, this subsection introduces only the regression model applied in this research: linear regression and random forest regression.

A linear regression model involves a linear combination of the input variables:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D. \tag{2.14}$$

The training process would take input features $\mathbf{x}$ and the output observations $y$ and minimise the loss function, for example, the mean square error (MSE) $= (y - \hat{y})^2$. The training process would optimise the weight value $\mathbf{w}$, and therefore, the model can be used for prediction. Linear regression is the basic regression model, with the strong assumption that the relation between $\mathbf{x}$ and $y$ is linear. To model non linear relations in general, one of the most effective machine learning models, random forest regression can be applied [Breiman, 2001b]. Random forest regression is a *combining model* or *ensemble learning model* which is a combination of multiple learning algorithms aiming at improving performance compared to linear models [Bishop and Mitchell, 2014]. A random forest is constructed by multiple decision trees. A decision tree is an algorithm that uses a sequence of binary selections corresponding to the traversal of a tree structure. This algorithm can be used for both classification and regression tasks. Having a "forest" instead of a "tree" is introduced to solve the overfitting problem for a single decision tree. An illustration diagram for the algorithm is provided in Figure 2.8. The algorithm firstly selects a random sample with replacement of the training set and fits trees to these samples, i.e. bagging. The tree

structure is generated and the split criteria is optimised using a loss function. Figure 2.8 shows a general structure of the forest. The nodes of the tree are trained to split based on one or several features. If after the splits the node only contains one category, this node would be labelled the leaf node and would stop splitting further. After training, the inputs propagate through the tree structure and fall into one node. This node will decide which category it is classified as. The random forest algorithm uses several trees, so the final result is the majority of the outcome of each tree. The green and red nodes in Figure 2.8 can represent the classes of the classification problem. In a regression case, the average of the output of each tree can be used as the final prediction.

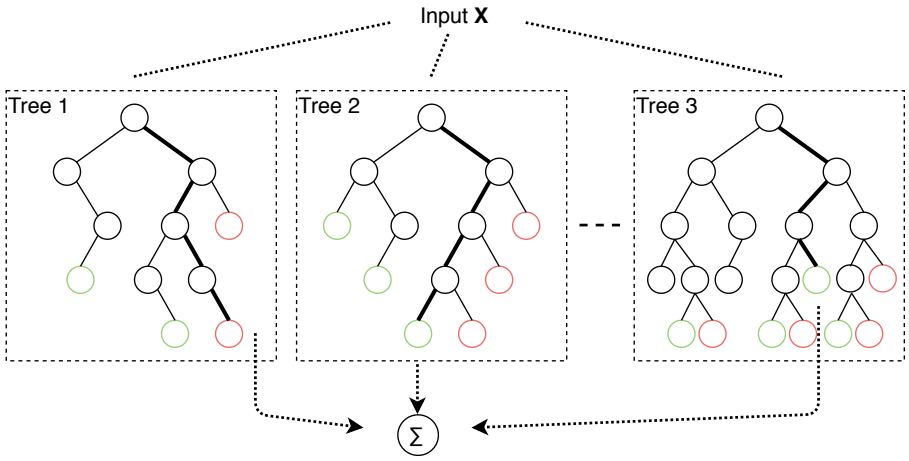



Figure 2.8: A random forest algorithm diagram

### 2.5.2 Feature Selection

Feature selection is a commonly used data preprocessing technique for optimisation of a machine learning model. By selecting the optimal features using large amounts of data, the number of features can be reduced, irrelevant or redundant features can be removed and computational cost will be reduced. The system is expected to become more robust to noisy data. This brings immediate benefits for applications: speeding up data mining algorithms and improve mining performance such as predictive accuracy and result comprehensibility [Baeza-Yates et al., 1999, Liu and Yu, 2005]. The general steps of feature selection are

- *subset generation;*
- *subset evaluation;*
- *selecting a stopping criterion;*

- *result validation*

in a sequence. When classifying the methods by subset generation, the strategies can be identified into these categories: 1) complete search, 2) sequential search or 3) random search. They represent the different ways of how to generate the subset from the full feature set. At subset evaluation stage, feature selection algorithms can be classified as *filter* model, *wrapper* model, and *hybrid* model.

A *filter* model is a model that evaluates the relations across features. It will select the features that contain the least shared information among them. It is not taking the performance into account. The selected feature set may contain the most information, but it is not guaranteed to be the most related to the training target. A *wrapper* model considers only the algorithm performance, which means it selects the best performing subset, but it can be easily overfitted. To solve this problem, a third type of model has been proposed. This is called hybrid model, which combines the filter and wrapper model. The stopping criterion is normally defined as the search being complete, or reaching a certain given bound such as the number of iterations. The most common result validation method is the direct use of the machine learning model performance.

In terms of audio feature selection, Doraisamy et al. [2008] considered several correlation based filter models, while in [Baume et al., 2014], researchers applied a wrapper model. There are embedded methods which combine selection strategy with machine learning algorithms. Since random forest regression is used in this Thesis, the author will also consider several feature ranking algorithms specific to random forest. Originally proposed in [Genuer et al., 2010] and implemented in [Ronan et al., 2015, Martínez Ramírez and Reiss, 2017], the feature importance method measures the change in the out-of-bag (OOB) error rate for each individual tree when replacing a certain feature with random values. The average performance change can be used as a measure of feature significance. In addition, another commonly used method described in [Stone et al., 1984] is *mean decrease impurity*. This is defined as the total decrease in node impurity averaged over all trees of an ensemble. Node impurity is a measure of the homogeneity of the labels in a node, for example a tree node in Random Forest algorithm. It can be approximated by the proportion of samples reaching a certain node.

### 2.5.3 Deep Neural Network

Neural networks are a set of algorithms inspired by the human brain's neural connections. The idea of neural network can be traced back to the 1940s. The invention of backpropagation and the development of GPUs and distributed computing helped these algorithms thrive in the 1980s and after 2010s. An artificial neuron is a computational node, which has linear weights and bias along with a non linear activation function. With a vast amount of artificial neurons connected, the model is able to learn a highly sophisticated function. Schmidhuber [2015] provides a good overview of the history, development and techniques of neural networks until 2014.

A node represents a scalar value as $X$ in Figure 2.9. An activation function is normally applied to each node as $f(z)$. A layer consists of a set of nodes representing a vector. They are not inter-connected, but connected by weight vector $w$ to the subsequent layer as $z = \sum(wx)$. The bias is treated as $w_0$ in the equation. In a network with multiple layers, there are intermediate layers as well as the input and the output layer. The intermediate layer can also be referred as hidden layers.

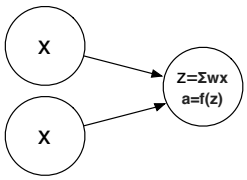



Figure 2.9: An example of nodes and the connections in a neural network.

#### Convolutional Neural Networks for Audio

A conventional DNN takes a single feature vector as input. In audio recognition tasks however, the objective is often to learn increasingly complex patterns that unfold in frequency and/or time. For this reason, convolutional networks that look at a region of some time-frequency representation such as STFT or learn temporal patterns from time-domain audio signals may be used. A convolutional layer in neural network normally uses a fixed size kernel to sweep over an input. The same weights (convolutional kernels) are applied to the whole input area. This results in vastly reducing the number of trainable parameters. The output is a representation of local activations of patterns. It has been widely used in 2D signals, e.g. image signals. It has shown an encouraging performance in audio domain

as well.

As it is discussed in the Chapter 1, DNN has an increasing trend in audio signal processing. It has shown outstanding performance in speech recognition [Palaz et al., 2015], music genre classification [Li et al., 2010], onset event detection [Schlüter and Böck, 2013], etc. There are attempts that use generative model to synthesise audio with certain effects. Most of these research use one DNN model to represent one effect with a specific parameter settings. Controlling the parameter configuration by generative model is not yet possible, c.f. Section 2.2.2. This research uses neural network to learn a feature embedding that would be used for parameter prediction rather than using it as an end-to-end model.

There are two ways to use CNN for audio related tasks. Since CNN is originally designed for image signal, i.e. 2D signal per channel, it is straightforward to transform the 1D audio signal into a 2D time-frequency representation and apply 2D CNN directly. Popular representation includes spectrogram [Pons et al., 2017c] and Mel-spectrogram [Choi et al., 2016, Ullrich et al., 2014]. Whether time-frequency representation of audio can be considered equivalent to an image remains a question, even it is proved to be powerful for tasks like classification and music tagging. For instance, images can be shifted in any axis and the information remains the same, but a shift in frequency axis will be different for audio, for example, it can result in perceived pitch shifting and potentially other effects. Many research shows that given raw audio input, it is possible for the model to learn an appropriate hierarchical representation [Cakır and Virtanen, 2018]. Therefore, to reduce information loss during the preprocessing, many recent research applied raw audio as the input for a CNN model. Normally raw audio is fed into the network as relatively long sequences of time-domain samples forming a large 1D input vector. Many researchers use comparatively large convolution filters, e.g. 10-20ms (441-882 samples if the sample rate is 44100Hz) [Ardila et al., 2016, Dieleman and Schrauwen, 2014]. There are also "sample level" networks which apply small filters, e.g. 3 samples Lee et al. [2017] and achieve a comparable level of accuracy in the state-of-the-art music tagging tasks. Since this research would require a model to learn multiple parameters of an audio signal processing task, the model needs to capture features at different scales. The multi-kernel model [Dieleman and Schrauwen, 2013, Pons et al., 2017b] designed to enhance the versatility of the model would be a benefit for model design.

There are a great amount of network architectures in the research area of deep learning. It is not feasible to cover them all in this section. Apart from CNN, one architecture

this Thesis applies is *Residual Block*. The intuition behind the Residual Block can be represented as follows: a normal block operates as $y = F(x)$, and a residual one operates as $y = F(x) + x$. The Residual Block was proposed to solve the problem of vanishing gradient [He et al., 2016]. It is a case where as the network gets deeper and therefore more complex, the gradient becomes so small that the numerical precision of the processor is unable to represent it and the value becomes zero and make optimisation, which relies on gradient, impossible. Good performance as a result of Residual Blocks make very deep network possible.

**Siamese Model**

The problem this research aims to solve requires the model to pay attention to subtle changes in an audio signal, such as changes in note attack times and learn features related to them. Therefore, the author considers to use a siamese model structure [Bromley et al., 1994]. The siamese model is a network structure that conceptually contains two or more identical subnetworks with shared weights. Figure 2.10 illustrates the structure, where the two branches have shared weights. These branches can be merged in the later stage of the network and followed by more neural network structures.

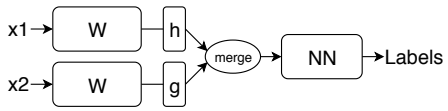



Figure 2.10: An example of siamese model.

The Siamese model is an appropriate structure when a model needs more than one input or branch, and all inputs are from the same domain. This structure is powerful especially when the multiple inputs are similar or linked in a certain relationship [Yu et al., 2016]. The siamese model is firstly proposed to target "one-shot" learning, where training example for one class is very limited [Koch, 2015]. This model will naturally rank the inputs by similarity, which is ideal for the problem this Thesis is focusing on. In this research, the inputs are linked by compression, i.e. the audio inputs differ only in their dynamic range related characteristics, which make it possible to use siamese model to learn feature embeddings that characterise these differences well. Similar structures are also used in other audio applications and it is shown to be useful for feature learning [Yang et al., 2018].

## 2.6 Conclusion

In this chapter, the author reviewed the related research within the topics of intelligent production, audio signal processing, audio similarity and auditory system, and machine learning. Relying on many previous works build the solid foundations of this research.

This chapter started with an introduction of audio effects with an emphasis on DRC. The author introduces the basic mechanism of this audio effect with explanations of its parameters. An overview of how producers and artists normally apply DRC is provided as well. It is followed by the introduction of adaptive audio effects.

Related research in a broad sense of intelligent production are introduced as well. The author provides a general overview, and compares the related applications. The audio features and signal processing algorithms are outlined in Section 2.3.

Since this research includes an audio similarity estimation method for evaluation purposes, an overview of audio similarity applications are provided in Section 2.4. This section also includes an introduction of the human auditory system and models for simulating the behaviour of the auditory system.

Section 2.5 reviews related machine learning models that are used in this research. It starts with the introduction of the typical regression models. Since conventional machine learning models require the selection of efficient feature, the author also introduces feature selection algorithms. The audio features are extended from handcrafted features to DNN feature learning. The review of related model structures is included in this chapter as well.

Having the support of these previous work, from the next chapter onwards, this Thesis will start to discuss a series of system design, feature design, and evaluation mechanisms. Experiments and evaluation for each proposed system component and method are included as well.

# Chapter 3

# Intelligent Control of Audio Effects

Digital audio effects are essential for all processes in music post-production. Since almost all music consumed by audiences are polished, a good mix may be taken for granted. However, professional quality sound requires substantial expertise, experience, time and labour to achieve. This Thesis acknowledges the substantial experience and craft of established audio engineers and producers. The approach proposed in this Thesis is also designed to improve and smooth the procedure for both professionals and amateurs. In this chapter, the proposed intelligent system is introduced in detail. This discussion starts with the motivation of the design, and define the main problem the author is aiming to tackle. The overall system design and the research schematic for individual research stages are also provided. This chapter focuses on describing workflows. It introduces the basic design for the functional parts of this research. For example, the feature extractor and audio similarity model will be improved in the following chapters. The system is evaluated on simple audio examples. With the improvement of each part of the system, specific components of our model will be evaluated with more complex audio examples in the following chapters.

## 3.1 Motivation

This section is started by discussing some motivating examples for automating audio effect configuration. Controlling effects requires significant experience and know-how, especially when used for aesthetic purposes during music production [McGrath et al., 2016]. This

often involves mapping a concept or idea concerning sound qualities to low-level signal processing parameters with limited meaning from a musical perspective. For instance, they often describe a desired effect by providing an example, e.g., name a particular style, artist or song, instead of a specific audio feature. Knowledge of signal processing, which was requisite for engineers in early studios, as well as good understanding of their control parameters and function constitute the skills of sound engineers and producers. Acquiring these skills however present a high barrier to musicians and casual users in applying today's production tools as discussed in Chapter 1. Consequently, the development of intelligent tools, as it has been done in other content production industries such as desktop publishing or word processing, may greatly benefit music production too.

Substantial amount of works in this area are concerned with automating the mixing and mastering process (see e.g. Reiss [2011] or Ma et al. [2015]). The approach proposed in this Thesis is significantly different from previous studies in that it does not directly target multitrack mixing and mastering, or attempt to use high-level semantic descriptors to control effects [Wilmering et al., 2012, Cartwright and Pardo, 2013]. The focus of this work is on the novel task of estimating the parameters of audio effects given a sound example, such that the processed audio sounds similar in some relevant perceptual attributes (e.g. timbre or dynamics) to the reference sound. This has applications in various stages of music production. For instance, while creating an initial rough mix of a track, artists may describe how they would like an instrument to sound using an actual sound example [McGrath et al., 2016]. An intelligent tool that provides audio effects settings based on a reference audio track is useful to meet this requirement. It may also help hobbyists and amateurs to make their own music or create remixes, an activity encouraged by well-known bands, such as Radiohead, by releasing stems and multitrack recordings.

As mentioned in Chapter 1, an innovative control mechanism is proposed in this Thesis by using a reference audio example. To investigate this idea, a single effect is selected: Dynamic Range Compression as a research subject for system design and evaluation. This Thesis focuses on this particular effect due to its several stages of non linearity within the typical signal processing algorithms involved in its implementation as well as its wide usage in music mixing and production c.f. Section 2.3.2. This chapter focuses on the introduction of the workflow and system design, therefore, the audio materials are kept simple: mono-timbral notes. Since the problem to be solved is to learn a parameter setting that brings the input audio closer to the reference, the system is designed in a way that

the machine learning model is fitted to the difference between the input and the reference. The proposed solution consists of

1. an audio feature extractor that generates features corresponding to each parameter,
2. a regression model that maps audio features to audio effect parameters,
3. a music similarity measure to compare the processed and the reference audio

The details are provided in the following section.

## 3.2 System Design

The system details are presented in this section. Section 3.2.1 provides the system workflow. The initial approach to this research introduces a simple feature extractor using conventional audio features introduced in Chapter 2, such as crest factor or spectral centroid. The training method for the regression model is described in Section 3.2.2. The test mechanisms are also introduced in this section.

This study uses a single-channel, open-source dynamic range compressor developed in the SAFE project [Stables et al., 2014]. It is a downward compressor with an amplitude detector. In the interest of brevity, the author will not discuss the operation of the compressor again in this chapter and assume the reader is familiar with relevant principles introduced in Section 2.1.3. Further discussion can be found in Zölzer et al. [2002], Giannoulis et al. [2013].

### 3.2.1 System Design and Workflow

There are several ways to frame the research problem, generate training data and evaluate the feature extraction and machine learning algorithms proposed in this Thesis. A high-level overview of the proposed method is shown in Figure 3.1. The purpose of the system is to make the *Output* audio, which is a processed version of the *Input*, sound as close as possible to the second input *Reference*. A vector of low level features related to each specific compressor parameter threshold ($\theta$), ratio ($\gamma$), attack time ($\tau_a$) and release time ($\tau_r$), i.e. $\boldsymbol{\rho} = \{\theta, \gamma, \tau_a, \tau_r\}$ are extracted from both *Input* and *Reference*. They are served as training features for the random forest regression model. The approach requires selecting or designing relevant audio features and training a machine learning model that maps the difference between reference and input audio features to audio effect parameters. In a real

world application of the proposed system, the reference audio would be different from the input audio, making the feature design more challenging. Therefore a simplification is introduced first, assuming that the difference between reference and target sound is only the application of the audio effect, i.e. the audio content to be processed is the same as the reference, except with a difference in dynamic range compression. This restriction is relaxed in later research, however it is useful as large amounts of training data can be generated automatically.

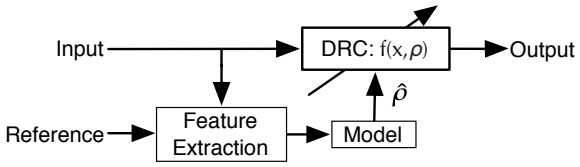



Figure 3.1: System overview

A thorough research schematic diagram is proposed in Figure 3.2. This diagram is aiming at abstracting the research problems this Thesis covers. It aims to introduce several problems that may be framed around the core challenges of this research, including possible methods of evaluation. The reader should note that this Thesis only covers a subset of the approaches proposed in the diagram. The description of each procedure is a high level recapitulation of the experiments in this chapter. Detailed description, evaluation, and results of each experiment will be followed in Section 3.2.2.

In the diagram, the nodes represent the audio materials used and generated in this research. Different combination represents different research process and problems. The details are as follows:

- I. Training procedure: Node 1, 2. Datasets are generated by manually controlling parameters $\boldsymbol{\rho}_1$. Most of the training datasets in this research are generated in this way - with controlled parameter settings, and including both processed and original audio. The training process normally uses this dataset and the ground truth parameter settings to train a regressor. This process of the research, i.e. the details of the training procedure is demonstrated in Section 3.2.2 - 1.

- II. Test procedure 1 (model evaluation, numerical prediction accuracy test): Node 3, 4, 5. In this case, an audio $B$, which is different from the training examples, is used as *Input*. The assumption is that the reference, $B''$ is generated from the input $B$ by

$\boldsymbol{\rho} = \{\theta, \gamma, \tau_a, \tau_r\}$

$Model$: trained model using selected audio features.

$f()$ : Compressor process function.

$E()$: extraction function for selected audio features.

$P()$: prediction function parameters $\boldsymbol{\rho}$.

$D()$: distance measurement for audio examples.

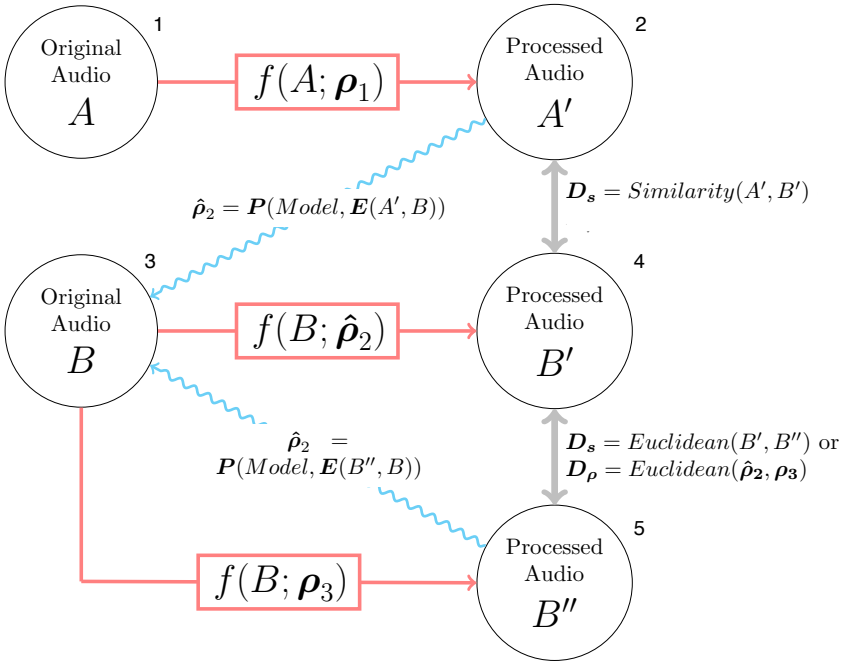



Figure 3.2: Overall schematic diagram

$\boldsymbol{\rho}_3$. Having the trained $Model$, it is able to predict this parameter. The prediction is denoted as $\hat{\boldsymbol{\rho}}_2$. The distance between $\hat{\boldsymbol{\rho}}_2$ and $\boldsymbol{\rho}_3$ can be measured as the numerical test, i.e. $\mathbf{D}_\rho$. Details of this process are given in Section 3.2.2 - 3. The value of the parameters do not represent audio perception. In this situation, the Euclidean distance or similarity between $B'$ and $B''$, i.e. $\mathbf{D}_s$ can be calculated as well.

- III. Test procedure 2 (similarity assessment): Node 1, 2, 3, 4. In this case, the *Reference* is $A'$. This reference is not generated from the *Input* $B'$. In this process, it is assumed that the origin of $A'$ is available to help with the prediction. This test is detailed in Section 3.2.2 - 4. After predicting $\hat{\boldsymbol{\rho}}_2$, $B'$ can be generated and perform evaluation. The low level features and the similarity between $A'$ and $B'$ can be compared, i.e. $\mathbf{D}_s$.

- IV. Test procedure 3 (similarity assessment in a realistic scenario): Node 2, 3, 4. A

more realistic situation is that the users only have the processed audio $A'$ instead of both $A$ and $A'$. The same evaluation as III can be performed in this procedure as well. Details are given in Section 3.2.2 - 5.

- V. Test procedure 4: Node 2, 3, 4, 5. Assuming it is possible to get the parameter setting $\boldsymbol{\rho}_3$ by either professional audio engineers or other sources, under the requirement of making $B''$ as close as possible to $A'$. In this case, *Reference* is $A'$, the output is $B'$ and $B''$ can be considered as the ground truth. Refinement and optimisation can be conducted using the similarity distance between $A'$ and $B'$. The same evaluation procedure as described in II can be accomplished. One possible optimisation mechanism will be detailed in Chapter 7.

This diagram helps to outline many design aspects of the proposed system. The parameters and process function of DRC are decided out of the training process. The regression model needs to be trained, which directly links to the prediction function $P()$. The feature extractor $E()$, the distance measure $D()$ need to be devised too. More details about the feature extraction stage will be provided in Chapter 4 and Chapter 5. The distance measurement function can be in different complexity level. In this chapter, 1D Euclidian distance as well as an audio similarity model from literature is used. An advanced audio similarity model designed to model differences in audio characteristics related to dynamic range compression is introduced in Chapter 6.

The following sections describe the first (proof of concept) experiment, with several simplifying assumptions on the type of audio material and the use case conditions of the proposed system.

### 3.2.2 Training and Testing Procedures

#### Regression model training

This section outlines the datasets and training procedure for the regression models that are used in the first experiment to map features to effect parameter settings. In the first stage of this research, the author considers two types of instruments: snare drum and violin. The former is one of the most common instruments that requires at least a light compression to even out dynamics. The drum samples are typically short and exhibit only the attack and release (AR) part of the typical sound energy envelop, c.f. Section 2.1.3. The violin recordings typically consist of a long note with fairly clear attack, decay, sustain and release

(ADSR) phase. All audio samples in this chapter are taken from the RWC isolated note database [Goto et al., 2003].

Table 3.1 describes the four violin note datasets denoted $A, ..., D$ that are used for training. In each dataset, one parameter of the effect is varied while the others are kept constant. The number of training samples in each dataset equals to the number of notes, i.e., 60 in case of the violin dataset, times the number of grid points (subdivisions) for each changing parameter. In this study, 50 settings are used for threshold and ratio, and 100 settings for attack and release time as it is shown in the first column of Table 3.1. The same process is applied to 12 snare drum samples to form the drum datasets. Each training set $A, ..., D$ is used for predicting a specific parameter.

| Training sets (size) | Conditions | | | |
|---|---|---|---|---|
| | $\theta$(dB) | $\gamma$ | $\tau_a$(ms) | $\tau_r$(ms) |
| A (60*50) | 0:1:49 | 2 | 5 | 200 |
| B (60*50) | 37.5 | 1:0.4:20 | 5 | 200 |
| C (60*100) | 37.5 | 2 | 1:1:100 | 200 |
| D (60*100) | 37.5 | 2 | 5 | 10:10:1000 |

Table 3.1: Training set generation details for violin notes

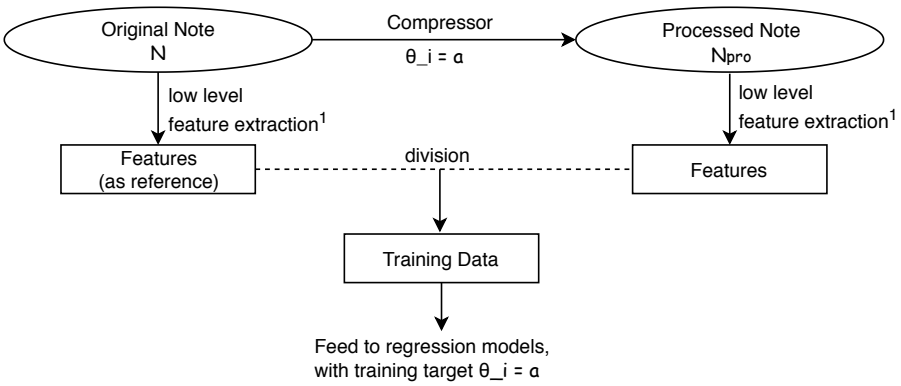



Figure 3.3: Flowchart of training data generation

Figure 3.3 describes the training process. Taking training set A as an example, the original notes $N$ are the recorded violin notes. The processed notes, denoted $N_{pro}$ on the right hand side of the figure, are generated from $N$, which are processed by the compressor with different threshold values. There are 60 different notes $N$, and each $N$ generates 50 processed notes $N_{pro}$. This yields $60 \times 50 = 3000$ $N_{pro}$ in the training dataset for threshold. Because the features from $N_{pro}$ are highly correlated with the original note

$N$, the difference between these two audio is used to focus on how the features change as a result of dynamic range compression. Therefore, it is the calculated ratio value this research actually uses to train the regression model. There are $l$ features related to the threshold extracted from each note. The conventional audio features used in training is listed in the next subsection. The detailed list of features will be provided in Chapter 4. Training data $A$ comprises 3000 $N_{pro} \times l$ feature vectors. In the following, this training data is used to train the regression model. The same principle applies to training sets $B$, $C$ and $D$.

At this stage of the research, two regression models are compared and evaluated: linear regression, and random forest regression [Breiman, 2001b]. Random forest uses averaging over sub-samples from the dataset to improve the prediction accuracy of the model as well as to mitigate over-fitting problems. The use of this latter model is motivated by the hypothesis that the relationship between the audio features and the compressor parameters may not be modelled accurately enough with simple linear regression due to the non linearities in the process. In the evaluation, the implementations of the machine learning models are available in the scikit-learn python module [Pedregosa et al., 2011]. The theoretical background and more details about the machine learning models used in this study are introduced in Chapter 2, Section 2.5.

**Feature extraction**

The experiments described in this chapter require audio features that are targeting each individual effect parameter. There will be some features shared by all parameters. Since DRC affects the dynamics of the signal, for instance, statistical features can be selected for all four parameters. By statistical features, the author means to extract audio features frame-wise, and then calculate the statistical features based on the frame-wise vectors. In terms of audio features, the author considers the perceptual attributes that are affected by the DRC, loudness and timbre [Wilmering et al., 2013]. The RMS features in the feature set reflect energy, which are related to loudness, while the spectral features reflect the spectral envelope, which is related to timbre. The statistical features are calculated frame-wise, with a frame size of 1024 samples and a 50% overlap. For spectral features, the spectrogram is calculated using 40 frequency bins up to 22kHz. It can be assumed that this bandwidth is sufficient for the control of selected DRC parameters.

The magnitude spectrogram is defined as $Y(n,k) = |X(n,k)|$ with $n \in [0 : N-1]$ and

$k \in [0 : K]$ where $N$ is the number of frames and $k$ is the frequency index of the spectrogram of the input audio signal with a window length of $M = 2(K + 1)$. The spectral features are extracted as described in Equations 3.1 - 3.4:

$$\mathsf{SC}_{\mathsf{mean}} = E\left[\frac{\sum_{k=0}^{K-1} k \cdot Y(n,k)}{\sum_{k=0}^{K-1} Y(n,k)}\right], \tag{3.1}$$

$$\mathsf{SC}_{\mathsf{var}} = Var\left[\frac{\sum_{k=0}^{K-1} k \cdot Y(n,k)}{\sum_{k=0}^{K-1} Y(n,k)}\right], \tag{3.2}$$

$$\mathsf{SV}_{\mathsf{mean}} = E[(E[Y(n,k)^2] - (E[Y(n,k)])^2)^{1/2}], \tag{3.3}$$

$$\mathsf{SV}_{\mathsf{var}} = Var[(E[Y(n,k)^2] - (E[Y(n,k)])^2)^{1/2}], \tag{3.4}$$

where SC stands for Spectral Centroid, and SV stands for Spectral Variance. The mean and variance of SC and SV in the equations are calculated across all $M$ length frames.

The following temporal RMS features are also extracted as described in Equations 4.3 - 4.4:

$$\mathsf{RMS}_{\mathsf{mean}} = E[(\frac{1}{M}\sum_{m=0}^{M-1} x(m)^2)^{1/2}], \tag{3.5}$$

$$\mathsf{RMS}_{\mathsf{var}} = Var[(\frac{1}{N}\sum_{m=0}^{N-1} x(m)^2)^{1/2}], \tag{3.6}$$

where $x(m)$ represents the magnitude of audio sample $m$ within each $M$ length frame, and the mean and variance are calculated across all the $N$ time frames as with the previous spectral features.

Four types of time domain features related to the attack and release of the notes as well as the speed of the compressor are also extracted. These are designed for isolated notes primarily or non-overlapping sound events. The attack and release times $T_A = T_{endA} - T_{startA}$ and $T_R = T_{endR} - T_{startR}$ are calculated using the RMS envelope through

a fixed threshold method (c.f. Peeters [2004]) that determines start and end times of the attack and release parts of the sound. Note that the attack and release of the notes are different from the attack and release time ($\tau_a$, $\tau_r$) as the DRC parameters. The end of the attack is considered to be the first peak that exceeds 50% of the maximum RMS energy. The RMS curve is smoothed by a low-pass filter with a normalised cut-off frequency of 0.47 rad/s. The RMS amplitude at the end of the attack and the start of the release are also extracted, i.e., $rms(T_{endA})$ and $rms(T_{startR})$ respectively, as well as the mean amplitude during the attack and release parts of the sound.

$$A_{att} = \frac{1}{T_A} \sum_{n=T_{startA}}^{T_{endA}} rms(n), \tag{3.7}$$

$$A_{rel} = \frac{1}{T_R} \sum_{n=T_{startR}}^{T_{endR}} rms(n), \tag{3.8}$$

where $T_{start}$ and $T_{end}$ are indices of the start and end of the attack or release. At last, a feature related to how fast the compressor operates is calculated. Firstly, the ratio between the time-varying amplitudes of input or original sound and the reference sound is calculated $s(n) = rms_{ref}(n)/rms_{orig}(n)$. Then the amount of time for $s(n)$ to reach a certain value using a fixed threshold is calculated. The features described in this chapter are used to demonstrate a proof of concept system. A broader set of features are considered in Chapter 4. A feature selection scheme will also be investigated in Chapter 4 to optimise the use of audio features and improve the prediction accuracy. In addition to this, handcrafted features designed specifically to describe DRC related audio characteristics corresponding to the DRC parameters are presented in this research, for mono-instrument notes as well as higher complexity audio materials, c.f. Chapter 4.

#### Numerical accuracy evaluation

As described in Section 3.2.1, there are several stages of evaluation. In this section, a numerical test designed to evaluate the regression model will be demonstrated, corresponding to the Test procedure 1 in Figure 3.2. This evaluation aims only at verifying the basic idea behind the use of a reference sound (or note) to be approximated. This is not a realistic scenario, because it assumes the user have access to both the processed and unprocessed

(original) version of the sound which is used as reference. This is needed because the predictor variables, i.e., the input to the regression model is calculated as the ratio of the audio feature data extracted from these recordings. This requires access to all pairs of $(N, N_{pro})$ (see Figure 3.3) in the training data. This scenario is presented in Figure 3.4 consisting of two parts. The components within the dashed line box represent the actual control system for the compressor with three inputs: the input note $I$ to be processed, the reference note $R$ to be approximated, and its corresponding original note $N$ from the training set. The output note $O$ outside of the dashed line box will be used in the evaluation, where it is used to compare the similarity of the output and the reference. As it is mentioned in Section 3.2.2, the regression model is trained on the difference between features extracted from the processed and unprocessed audio pairs. When testing, the same data is needed to predict the compressor parameters. Therefore, the original note $N$ is provided in the process of generating feature vectors for testing.

Before using the similarity model depicted in the next subsections, e.g. the right hand side of Figure 3.4, this numerical test is performed to evaluate the regression model accuracy first. The experiment will compare predicted parameter values with the actual ones. The workflow is the same as Figure 3.3, providing a standard testing step for regression models. In this study, repeated random sub-sampling validation (Monte Carlo variation [Burman, 1989]) is applied to provide a more general model performance evaluation. The results and analysis are reported in Section 3.3.1.

**Similarity assessment**

Considering the motivations and use cases described previously, the desired output of this algorithm is to make an unrelated note $I$ sound similar to the reference note $R$, where $R$ is generated from $N$ through a compressor with e.g. its threshold set to $x$dB. However, even if the prediction is perfect with $x_p = x$, the same compressor for note $I$ and note $N$ can give different perceptual results. Therefore, a similarity model which takes this into account is needed to evaluate the similarity between $R$ and the algorithm output $O$. The processing and evaluation workflow is represented in Figure 3.4 with the structure within the dashed line box used to control the system while the components on the right hand side are used in the similarity test.

The similarity method used in this experiment is a simple and frequently used audio similarity model [Aucouturier and Pachet, 2002]. A simple audio feature which is a good

(although partial) indicator of the overall dynamics of the signal, i.e. crest factor (c.f. Section 2.3.2) is applied as well. The first stage of the experiment reports the crest factor difference between the reference $R$ and the output $O$. Secondly, the similarity between the two audio is measured using a Gaussian Mixture Model trained on Mel Frequency Cepstrum Coefficients (MFCC). Accordingly, the feature extraction in the workflow indicates the calculation of the divergence between two multiple Gaussian models, which provides the similarity information. The symmetrised divergence, Jensen - Shannon divergence (JS divergence) [Jensen et al., 2009], which is commonly used for Gaussian models, is applied. It is similer to KL divergence, but symmetrical and smooth. It is used to measure the similarity between two Gaussian distributions. The divergence between mixtures of Gaussian models is not analytically tractable, therefore the approach proposed in [Hershey and Olsen, 2007] is applied which is based on variational Bayes approximation. This model is a baseline technique that provides a proxy for timbre similarity. A further development of the similarity model is provided in Chapter 6. The results of this test and analyses are provided in Section 3.3.

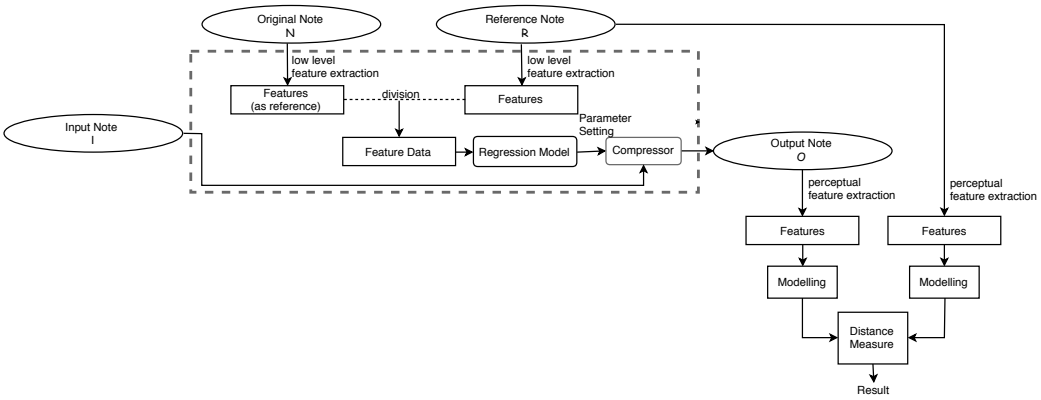



Figure 3.4: Workflow of initial system with access to the reference sound and its corresponding unprocessed version.

### Similarity assessment in a realistic scenario

This section repeats the previous experiments under different assumptions. In a real world scenario, if the reference audio $R$ is a commercial audio track, its corresponding unprocessed original sound $N$ is not likely to be available. In this condition, the proposed solution is outlined in Figure 3.5, where the input of the system is limited to the input note $I$ to be processed and the reference note $R$. In the feature computation workflow, the original note

$N$ is replaced by the input note $I$, because the features capture the difference between the reference note and the original note. Measuring the difference between the reference note $N$ and the input $I$ can be seen more reasonable and closer to a real world scenario. This process is corresponding to the Test procedure 3 in the overall diagram depicted in Figure 3.2. The evaluation of this system design is provided in the following section.

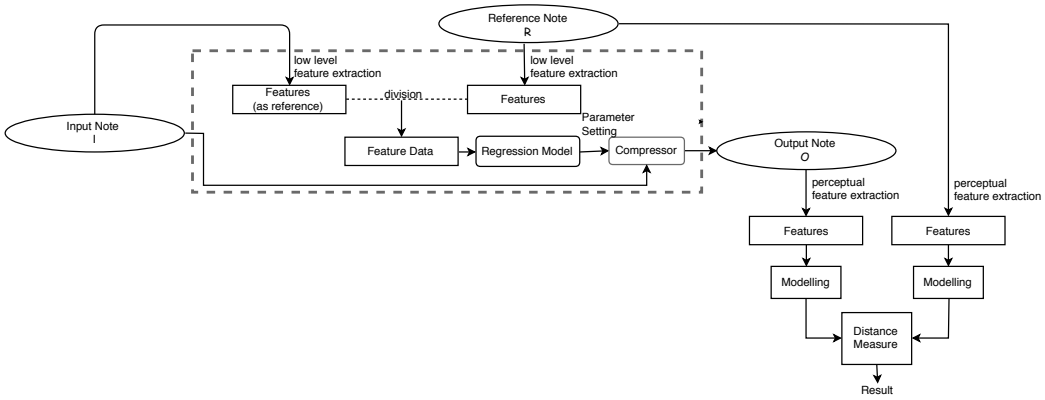



Figure 3.5: Realistic workflow with only the input and reference sound available.

## 3.3 Evaluation on Mono-instrument Notes

The previous section includes three stages of evaluation. The first stage is a numerical test and the results for this part are given in Section 3.3.1. The two stages of similarity tests are provided in Section 3.3.2. The audio materials used in these experiments are mono-instrument notes. All details are provided in Section 3.2.2.

### 3.3.1 Direct assessment of parameter estimation

Using the test procedure outlined in Section 3.2.2, here the author reports the accuracy of direct parameter estimation using random sub-sampling validation. Eighty percent of the data are used as training and twenty percent for testing. Two regression models are compared and evaluated: simple Linear Regression (LR) and Random Forest Regression (RF) [Breiman, 2001a]. Table 3.2 & 3.3 show the mean absolute errors for both instruments and regression models. Since the observed feature values are relatively small, we linearly scale the feature values to $[0, 1]$ and compare the errors. The highlighted values in the table show that the smallest error is always observed when using the scaled features and the random forest regression model. For completeness, the range for the four parameters

are (0,50] dB for threshold, [1,20] for ratio, (0,100] ms for attack time, and (0,1000] ms for release time. Scaling is reasonable in this pilot experiment, because the test data (reference) is selected randomly from the database, which means all $R$ has its original $N$ available. However, in a real world scenario, the reference sound is not taken from the database prepared to train the regression models. It is more likely to be a produced sound or track without access to its unprocessed version. Therefore the scaling factor of the training and test sets may differ. Scaling will not be used in the subsequent studies. Please note that there is no overfitting to the models, because in the evaluation the reference note and the corresponding original are excluded from the training set of the regression model.

The results also illustrate that the prediction accuracy for drums is better than violins in all cases. One reason might be that drum samples are shorter and exhibit a simpler structure - short sustain, followed by release and there is no pitched content. It shows that the system can predict the compressor parameters for drums better than for violins.

| **Violin** | | LR | RF |
|---|---|---|---|
| Threshold(dB) | error | 3.756 | 2.601 |
| | scaled features - error | 1.860 | **1.731** |
| Ratio | error | 2.065 | 1.583 |
| | scaled features - error | 0.110 | **0.091** |
| Attack(ms) | error | 15.503 | 0.719 |
| | scaled features - error | 1.012 | **0.686** |
| Release(ms) | error | 210.43 | 13.973 |
| | scaled features - error | 78.913 | **10.583** |

Table 3.2: Numerical test using linear and random forest regression model for violin notes

| **Snare Drum** | | LR | RF |
|---|---|---|---|
| Threshold(dB) | error | 1.185 | 0.800 |
| | scaled features - error | 0.408 | **0.345** |
| Ratio | error | 1.571 | 0.999 |
| | scaled features - error | 0.669 | **0.305** |
| Attack(ms) | error | 6.867 | 0.860 |
| | scaled features - error | 2.260 | **0.017** |
| Release(ms) | error | 40.960 | 6.851 |
| | scaled features - error | 23.045 | **0.999** |

Table 3.3: Numerical test using linear and random forest regression model for snare drum samples

### 3.3.2 Evaluation of similarity assessment between notes

The evaluation focuses on estimating the audio similarity in this section. The experiment is outlined in the right hand side of Figure 3.4 and Figure 3.5. Firstly, the crest factor is extracted, i.e., peak-to-RMS ratio as the similarity feature because it is correlated with the overall dynamic range of the signal. Based on the design, the crest factor of the reference note $R$ should be closer to the output note $O$ than the input note $I$. An example of this test is given in Figure 3.6 with 25 randomly picked test cases. The crest factor of the input signal is represented by the constant at the top of the figure and the crest factor of a series of reference notes are depicted by the blue curve at the bottom. The crest factor of the output signal from the system is shown in the middle (green curve). It is consistently brought closer to the reference which fits the expectation of this research.

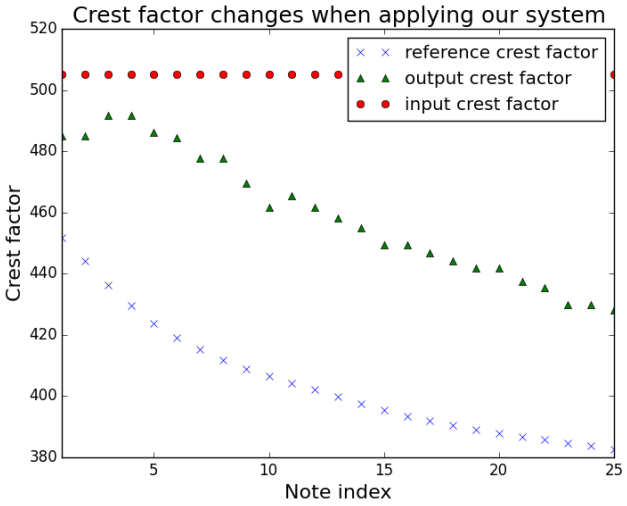



Figure 3.6: Example of changing crest factor with a fixed input and decaying reference sound

Additionally, 50 reference notes are tested and the results are presented in Table 3.4 for violin notes and Table 3.5 for snare drum, where $D_{Crest}(A, B) = mean(|Crest(A) - CrestB|)$. The results indicate that the system manages to bring the output closer to the reference using both regression models. In all parameters except threshold, the random forest model outperforms simple linear regression.

Next, this section discusses the results of similarity assessment as described in Section 3.2.2. At this stage, a simple audio similarity model is used to test the efficiency of the system. The procedure starts from extract MFCC coefficients as features using fixed-length overlapping time windows, and fit a GMM on the MFCC vectors. An approximation of

| **Violin** | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| $D_{Crest}(N_{pro}, R)$ | 60.31 | 94.13 | 104.93 | 85.31 |
| $D_{Crest}(N_{pro}, O)_{LR}$ | **12.53** | 39.72 | 46.76 | 48.62 |
| $D_{Crest}(N_{pro}, O)_{RF}$ | 15.27 | **38.24** | **45.23** | **47.19** |

Table 3.4: Average of Crest factor difference - Violin

| **Snare Drum** | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| $D_{Crest}(N_{pro}, R)$ | 49.14 | 70.04 | 50.47 | 70.68 |
| $D_{Crest}(N_{pro}, O)_{LR}$ | **27.99** | 43.85 | 27.28 | 44.63 |
| $D_{Crest}(N_{pro}, O)_{RF}$ | 29.33 | **43.45** | **27.20** | **43.44** |

Table 3.5: Average of Crest factor difference - Snare Drum

the symmetrised KL divergence is then calculated and used as a dissimilarity measure. Using the same procedure as in the previous part, the compressor settings provided by this algorithm should bring the output note $O$ closer to the $R$ compared to the input note $I$. Thus it is reasonable to assume that $D(R, I) > D(R, O)$ holds and the performance of the regression models can be tested. In this experiment, 50 original notes $N$ are selected and the reference $R$ are generated by changing one parameter 50 times at a time. The average similarity of the all $50 * 50 = 2500$ cases for each parameter and both regression models are demonstrated in Table 3.6 & 3.7. The distances between the output notes and the reference are closer to the ones between inputs and references. This result provides strong support for the assumption that since the similarity algorithm theoretically captures the timbre information as well, it will yield different results on different instruments. In this test, the distance reduction achieved by the system is larger for violins, i.e., the violin notes exhibit better results than the snare drums. This is possibly due to the fact that the MFCC features used here do not model the drum sounds well enough.

Finally, the author investigates how the proposed algorithm works in a more realistic scenario. When the original note $N$ is not available, it is reasonable to use the input note $I$ in place of $N$. Under this condition, the same test for both crest factor and the MFCC-based similarity model are repeated. The results for crest factor are provided in Table 3.8 for violin and in Table 3.9 for snare drum samples. The system is still able to bring the crest factor of the output $O$ closer to the reference $R$, but the efficiency is worse compared to the case when the original note is available. Random forest regression still yields better performance in almost all cases.

The result using the MFCC-based similarity model is illustrated in Figure 3.7 & 3.8.

| **Violin** | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| $D(R, I)$ | 38.122 | 53.187 | 44.018 | 55.206 |
| $D(R, O)_{LR}$ | 19.799 | 20.911 | 22.852 | 20.994 |
| $D(R, O)_{RF}$ | **19.742** | **20.856** | **22.213** | **20.807** |

Table 3.6: KL Divergences for the first workflow in Figure 3.4 - Violin

| **Snare Drum** | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| $D(R, I)$ | 77.497 | 112.368 | 73.559 | 91.487 |
| $D(R, I)_{LR}$ | 73.749 | 88.574 | 73.307 | **85.022** |
| $D(R, I)_{RF}$ | **73.696** | **88.487** | **73.238** | 86.081 |

Table 3.7: KL Divergences for the first workflow in Figure 3.4 - Drum

| **Violin** | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| $D_{Crest}(R, I)$ | 60.31 | 94.13 | 104.93 | 85.31 |
| $D_{Crest}(R, O)_{LR}$ | 34.86 | 29.37 | 68.74 | 75.94 |
| $D_{Crest}(R, O)_{RF}$ | **30.11** | **25.36** | **67.82** | **54.02** |

Table 3.8: Average of Crest factor difference - Violin

| **Snare Drum** | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| $D_{Crest}(R, I)$ | 49.14 | 70.04 | 50.47 | 70.68 |
| $D_{Crest}(R, O)_{LR}$ | **38.01** | 66.18 | **21.37** | 44.05 |
| $D_{Crest}(R, O)_{RF}$ | 42.90 | **45.24** | 27.37 | **42.75** |

Table 3.9: Average of Crest factor difference - Drum

In each subplot, $D(R, I)$ is on the left, $D(R, O)_{LR}$ is in the middle and $D(R, O)_{RF}$ is on the right. In Figure 3.7 for violin notes, the average divergence is not as promising, especially when comparing with the results in Table 3.6, but it is clear that even if the given reference sounds have a large diversity, the algorithm reduces this significantly, and shows a very stable improvement in the similarity result. On average, the random forest regression performs better than linear regression in all cases except when predicting threshold. This shows the benefit of modelling non linearities. Therefore the system will use random forest regression in the following experiments. Figure 3.8 shows that the output of the system did not manage to achieve a dramatic reduction in the similarity distance in case of the snare drum. As explained before, further investigation is needed for the influence of timbre on this similarity algorithm. Furthermore, larger datasets will be considered in subsequent experiments.

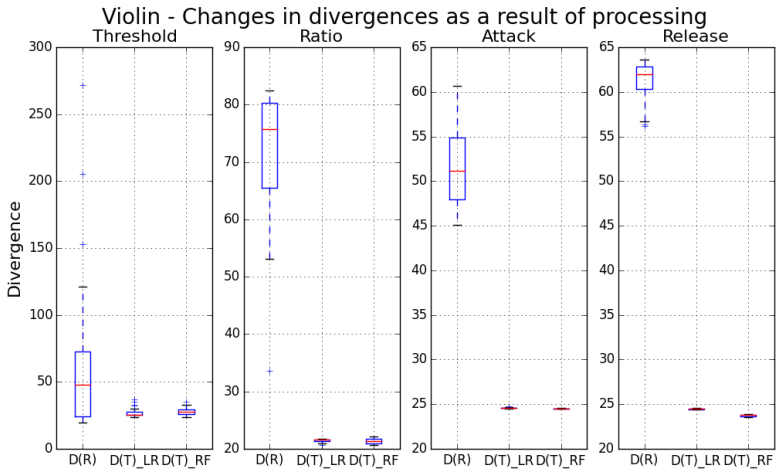



Figure 3.7: Similarity for four parameters in the second workflow, assuming the origin note N is not available. - Violin. The first column of each sub-figure is the distance between the reference and input, the second is the distance between the output and input using linear regression, and the third one is using random forest. D(R) is equivalent of D(R,I), and D(T) is equivalent of D(R,O).

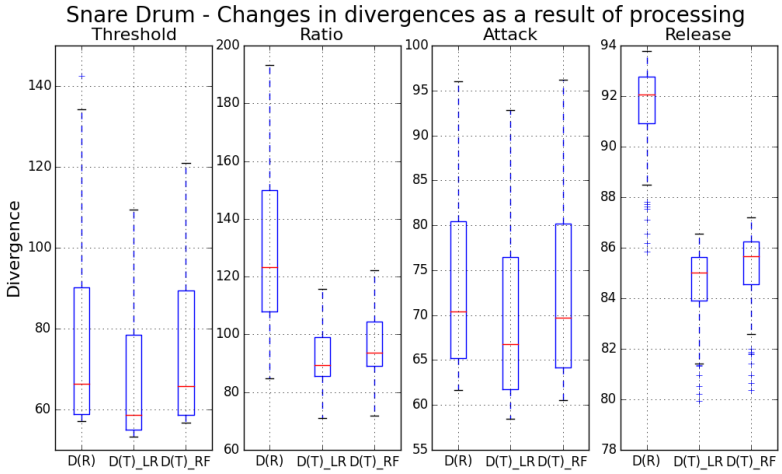



Figure 3.8: Similarity for four parameters in the second workflow, assuming the origin note N is not available. - Snare drum. Three columns of each sub-figure are the same as the previous figure. D(R) is equivalent of D(R,I), and D(T) is equivalent of D(R,O).

## 3.4 Conclusion

In this chapter, an innovative method to estimate dynamic range compressor parameter settings using a reference sound is proposed. The author demonstrates the first steps towards a system to configure audio effects using sound examples, with the potential to democratise the music production workflow. The progress from using a linear regression model to a random forest model is discussed. As DRC is a highly non linear audio effect, random forest regression shows better performance in almost all the evaluation cases. The evaluation progresses from a designated test case to a real world scenario as well. The results shows a promising trend in most cases and provides an initial indication of the utility of this approach. Drums samples also show better performance in almost all the cases. This may be due to the fact that the structure of drum samples are somewhat simpler than violin notes.

In the following chapters, the author will improve the feature sets by designing and selecting handcrafted features as well as building a feature learning model. The audio materials will be extended from simple mono-instrument notes to loops and polyphonic music tracks to fit the real world scenario better. In terms of evaluation, the similarity algorithm is improved using an approach that targets the audio perceptual aspects affected by DRC. A objective experiment related to audibility threshold of DRC parameters is carried out in Chapter 7 focussing on evaluation.

# Chapter 4

# Feature Design and Selection for Mono-instrument Loops

As described in the previous chapter, the proposed system has been divided into several components. In this chapter, the author will discuss further investigations aiming to improve the feature extractor component first. It can be assumed that using features with an emphasis on specific DRC parameters will improve the performance. Therefore, four feature sets are designed and selected for four DRC parameters. There are overlapping features in these feature sets, but there are also feature designed specifically for a single parameter. An extended set of features are tested in this chapter, including conventional features and handcrafted features specially designed for DRC parameter estimation. The previous chapter focuses on illustrating the system design, thus the audio materials used in the experiments are simple isolated notes. In this chapter, not only the features will be developed further and discussed in more detail, but the study is extended to a more complex scenario, mono-instrument loops. A *loop* is a short snippet of musical audio that may be tiled and repeated seamlessly to provide accompaniment. Some handcrafted features are designed based on the note structure, therefore, audio decomposition algorithms (see Section 2.3.3 in Chapter 2) are required when the objective is to separate note structures from a loop. In the second half of the chapter, several feature selection algorithms (see Section 2.5.2 in Chapter 2) are tested and applied on the full feature lists due to the possibility of redundancy in the feature set and to improve computational efficiency. Finally this chapter concludes with proposing an optimal set of features for each individual

compression parameter.

## 4.1 Motivation

To recap, this research is aiming to help with the labour intensive and time consuming problem of controlling audio effects. In the system design proposed in the previous chapter, a set of acoustic features are used to capture important characteristics of simple sound examples. These are then mapped to audio effect control parameters using regression. It can be assumed that the key to the performance of the proposed system is the efficiency of the feature extractor. It can also be assumed that designing specific handcrafted features will lead to improved performance. These considerations govern the research discussed in this chapter. Specific sets of features are proposed for each parameter. Audio decomposition algorithms are also applied for feature extraction on more complex audio materials.

In addition to the proposed feature sets in this and the previous chapter as already mentioned, there is a possibility that redundant information may exist, especially the possible overlap between the DRC parameter specific features and the conventional ones. Many machine learning research includes a feature selection step before training a prediction model. This step may be beneficial in this research as well. One reason for investigating this is to remove possible redundancies. Another reason is that the computational cost may drop if the prediction model works well with fewer features. It may also have the potential to improve the performance. This chapter includes several selection strategies to balance relevance, generality, and performance. It aims to find the optimal feature set using feature selection methods. The final set of features are settled by balancing the results from all the possible selection strategies.

## 4.2 Feature Sets

This section provides details about the feature set this work has used and developed. In this research, four DRC parameters are in the prime focus: threshold, ratio, attack and release time, i.e. $\boldsymbol{\rho} = \{\theta, \gamma, \tau_a, \tau_r\}$. These are the most commonly used parameters of the DRC. Other parameters, e.g. the make-up gain are not considered in this Thesis as it is a simple level shift operation. The standard frequency and time domain features partly introduced in Chapter 3 are used as universal features for the prediction of all parameters, as described in Subsection 4.2.1 and 4.2.2. The features designed specifically for $\tau_a, \tau_r, \gamma$,

are detailed in Section 4.2.3.

The feature design discussed in this chapter starts with the features applicable to isolated notes. These will be adapted to audio loops and more complex polyphonic material using audio decomposition techniques such as NMF.

### 4.2.1 Frequency Domain Features

Frequency domain statistical features are the most commonly used features for predicting DRC parameters [Ma et al., 2015, Zölzer et al., 2002] because the statistical features are strongly related to dynamics as explained in the Background Chapter (see Section 2.3.2). In this research, the author defines the magnitude spectrogram $Y(n,k) = |X(n,k)|$ with the notations of $n \in [0 : N-1]$ and $k \in [0 : K]$ where $n$ is the frame index and $k$ is the frequency index of the STFT, $X(n,k)$, of the input audio signal with a window length of $M = 2(K+1)$. Since the signal $Y(n,k)$ has two dimensions, the author uses two typefaces to distinguish the mean and variance over the temporal dimension and the frequency dimension. $\mathfrak{E}$ and $\mathfrak{Var}$ are used to represent the calculation over time, while $E$ and $Var$ are used across frequency. The mean and variance operations across both dimensions are aimed to capture most of the dynamic changes. The features related to the first order statistical frequency feature, spectral centroid, are given in Equation 4.1-4.2. The list is extended until the 4th moment, therefore, the frequency domain feature set is as follows: $SC_{mean}$, $SC_{var}$, $SV_{mean}$, $SV_{var}$, $SS_{mean}$, $SS_{var}$, $SK_{mean}$, $SK_{var}$, where SC stands for spectral centroid, SV stands for spectral variance, SS for spectral skewness, and SK for spectral kurtosis. They are calculated in the same way as the provided equations.

$$\mathsf{SC}_{\mathsf{mean}} = \mathfrak{E}\left(\frac{\sum_{k=0}^{K-1} k * Y(n,k)}{\sum_{k=0}^{K-1} Y(n,k)}\right), \tag{4.1}$$

$$\mathsf{SC}_{\mathsf{var}} = \mathfrak{Var}\left(\frac{\sum_{k=0}^{K-1} k * Y(n,k)}{\sum_{k=0}^{K-1} Y(n,k)}\right), \tag{4.2}$$

Additionally MFCC features are extracted. As the Cepstrum represents the envelope of Mel-scaled spectrograms, MFCC are commonly used to represent certain aspects of the timbre of an audio signal. Given frame-wise MFCCs, $M(n,k)$, with $k \in [0, 13]$ represents the first 13 Mel-frequency Cepstrum coefficients, and $n \in [0 : N-1]$ represents the index

of the time frame. Using $M(n,k)$ to replace $Y(n,k)$ in Equation 4.1-4.2, the statistical features can be obtained based on MFCCs too. It is safe to assume the higher order statistical frequency characteristics have already been included in the previous frequency domain features, therefore, only the mean and variance of the first two moments of MFCCs is used, i.e. $MC_{mean}$, $MC_{var}$, $MV_{mean}$, $MV_{var}$.

### 4.2.2 Temporal Features

Statistical features in the time domain are calculated in the same fashion as the frequency domain features. Unlike spectrograms, time domain audio samples are in one dimension. Therefore, this work calculates the mean and variance up to the second moment of $x(m)$, the magnitude of audio sample $m$ within each M-length frame. Therefore there will be $T1_{mean}$, $T1_{var}$, $T2_{mean}$, $T2_{var}$ as time domain features. Since many of the DRC interventions may happen over short time periods and work directly on the audio sample level, we are hoping these features could help to capture the dynamic characters changing at the sample level.

RMS features are considered as well using the RMS curves also with a window size of $M$. The mean and variance, $RMS_{mean}$ and $RMS_{var}$ across $N$ time frames which correspond to the average and variance of energy are also used as temporal features. The RMS features are calculated by Equation 4.3 - 4.4. The time domain features are calculated in the same fashion.

$$\mathsf{RMS}_{\mathsf{mean}} = E\left[\left(\frac{1}{M}\sum_{m=0}^{M-1} x(m)^2\right)^{1/2}\right], \tag{4.3}$$

$$\mathsf{RMS}_{\mathsf{var}} = Var\left[\left(\frac{1}{N}\sum_{m=0}^{N-1} x(m)^2\right)^{1/2}\right], \tag{4.4}$$

### 4.2.3 Features specific to DRC parameters

Although parameters are not working independently in the DRC process, it is still possible to design specific features that reflect the role of each parameter. In this section, the author introduces one feature for ratio and six features for attack and release times respectively.

The feature for ratio is the average of all the samples of which amplitudes are above the threshold, assuming there has been already a fairly accurate prediction of threshold

before predicting ratio. The amplitude reflects the compression ratio directly, except for the attack and release phases, where a smooth curve instead of the real ratio is applied.

$$\mathsf{R_a} = \frac{1}{M}\sum_{m=0}^{M-1}|x(m)|, \forall|x(m)| > threshold \tag{4.5}$$

Since the attack and release times are parameters that affect only a certain phase of the audio, the author proposes attack/release phase related features to improve the prediction. Equation 4.6-4.8 are the features representing the length, the average energy of the attack phase, and the energy at the end of the attack phase, where the attack time $T_A$ is calculated using the RMS envelope through a fixed thresholding method (c.f. Peeters [2004]). The end of the attack, $N_{endA}$, is considered to be the first peak that exceeds 90% of the maximum RMS energy and the start of the attack, $N_{startA}$, is the first sample of the RMS envelope that exceeds 10%. The RMS curve is smoothed by a low-pass filter with a normalised cut-off frequency of 0.47 rad/s.

$$\mathsf{T_A} = (N_{endA} - N_{startA})/Fs, \tag{4.6}$$

$$\mathsf{A1_{att}} = \frac{1}{N_{endA} - N_{startA}}\sum_{n=N_{startA}}^{N_{endA}} rms\_curve(n), \tag{4.7}$$

$$\mathsf{A2_{att}} = rms\_curve(N_{endA}), \tag{4.8}$$

**Procedure 1** Calculate $A3_{att}$

**Input:**

rms1 : non-compressed audio rms curve;
rms2 : compressed audio rms curve;

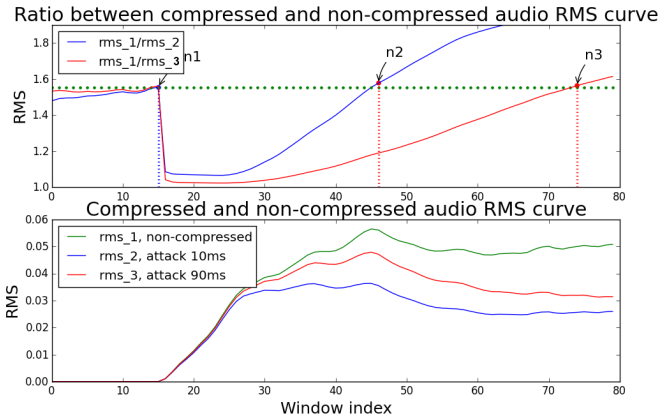
**Output:**

$A3_{att}$;

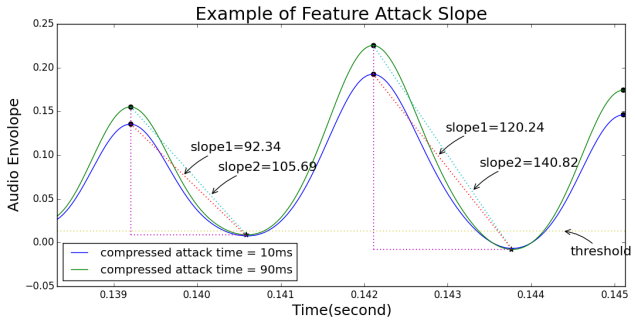
1: $\gamma$ = rms1/rms2
2: $n1 \rightarrow \forall\gamma[0 : n1] \leq 1.0$
3: $n2 \rightarrow \forall\gamma[n1 : n2] \leq 1.0$
4: $A3_{att} = n2 - n1$

Additionally $A3_{att}$ is calculated using the pseudocode shown in Procedure 1. For vi-

sualisation, an example is plotted in Figure 4.1(a). The ratio between the time-varying amplitudes of input or original sound and the reference sound is shown in the top figure. The ratio curve before intersect "n1" corresponds to the noise part before the actual audio content. Noise passed through a system will generate an arbitrary gain, so the start of the audio can be found through this ratio curve (intersect "n1") while it can also be used as a threshold to find when the ratio rises back to the threshold of interest (intersect "n2" and "n3"). The distance between the two dots clearly shows the speed of the operation of the compressor, where a short distance between "n2" and "n1" is corresponding to a small attack time, and the longer distance between intersect "n3" and intersect "n1" is for a longer attack time.




(a) Feature $A3_{att}$




(b) Feature $A4_{att}$, $A5_{att}$

Figure 4.1: Examples to demonstrate the procedure of generating attack time features

$A4_{att}, A5_{att}$ are calculated using the pseudocode in Procedure 2. During the transient part of the note, there are most likely several ripples. The slope of the ripples can reflect how fast the compressor operates, c.f. Figure 4.1(b). The RMS curve which has larger energy slope responds to a longer attack time. Based on this observation, the author designed $A4_{att}, A5_{att}$, which correspond to the mean and variance of the slopes. Features

**Procedure 2** Calculate $A4_{att}$, $A5_{att}$;

**Input:**

rms : the rms curve of the input audio;
threshold : the threshold of the compressor;

**Output:**

$A4_{att}$, $A5_{att}$.

rms1 $\leftarrow$ rms$[N_{startA} : N_{endA}]$
$\omega \in \Omega$, $\Omega$ = *peaks over threshold in sequence rms1,*
$\phi \in \Phi$, $\Phi$ = *notches after each peak,*
**for** $\omega_i \in \Omega$ **do**
    $s_i = (\omega_i - \phi_i)/dist(\omega_i - \phi_i)$
**end for**
$A4_{att} = \frac{1}{M}\sum_{i=0}^{M} s_i$
$A5_{att} = [\frac{1}{M}\sum_{j=0}^{M}(s_j - \frac{1}{M}\sum_{i=0}^{M} s_i)^2]^{1/2}$

corresponding to release time, $T_R$, $A1_{rel}$-$A5_{rel}$, are calculated in the same fashion but at the release phase.

Overall, 25 features are calculated for each note, of which 18 features are used for threshold, 19 for ratio and 24 for attack and release time. In the following section, these features will need to be adopted to more complex audio. The author will introduce several decomposition methods before extracting these features.

There is a possibility that the features contain redundancy. Especially the frequency domain features may not be able to reflect the change of attack and release times significantly, albeit it may be assumed that these temporal processes affect the spectral characteristics and/or perceived timbre of sounds. Therefore, feature relevance with respect to individual parameters will be assessed using different feature selection processes later in this chapter.

## 4.3 Audio Decomposition and Feature Design for Loops

As introduced earlier, this work uses standard audio features along with novel ones. Frame-wise spectral centroid, variance, RMS and many more are extracted and the mean and variance across the frame are used as standard features due to their relation to dynamics. DRC involves several stages of non linearity and has a different behaviour during transient and stationary parts of sounds due to the attack and release time parameters. The strategy is to decompose loops into simpler audio excerpts so that the attack and release phases can be measured more accurately. The workflow involves decomposing the audio into excerpts and then extract the features as listed before. The average of each excerpts will be used as the final features.

Three approaches are applied and examined including onset event detection, Non-negative Matrix Factorisation (NMF) and transient/stationary separation using Iterative Shrinkage Threshold Algorithm (ISTA).

The first and most straightforward method proposed here is using onset event detection to cut a long audio recording into short excerpts following the "note" structure. Guidelines for choosing the appropriate onset detection function can be found in [Bello et al., 2005]. More details are provided in the Background Chapter. In this section, the audio materials are mono-instrument loops, which are not highly complex material in the audio decomposition research area, therefore the commonly used, easy to implement and computationally light method, High Frequency Content (HFC) [Bello et al., 2005], is applied. The second approach is based on source separation using NMF to decompose complex audio into activation patterns. Finally, the author explores the transient/stationary audio separation algorithms to locate notes. The ISTA [Siedenburg and Doclo, 2017] is used for this purpose.

### 4.3.1 Onset event detection

The loops can be separated into notes using High Frequency Content (HFC) with a strong assumption of little overlap across adjacent notes. As outlined in Section 2.3.3, HFC is a relatively accurate method for onset event detection, and it is sufficient for the types of material investigated in this chapter. The author then applies feature extraction as described in Procedure 3. Several stages of selection schemes is applied to relax the assumption mentioned at the beginning of this paragraph. After obtaining onset positions, notes with attack/release phases that have not been smeared by other notes are selected using two conditions. Firstly, the notes that are longer than 1ms are kept. Shorter notes normally indicate significant overlap. The second condition is *goodness of fit* using two functions motivated by assumptions on the note envelope. A polynomial function fitted on the ascending part of the note envelope and an exponential decay function on the descending part. If the fitted parameters do not show the ascending/descending trend the note is discarded. The procedure secures that only the clear attack/release phases within the loops are selected. Features are calculated according to Equation 4.6-4.8 and averaged over selected notes. The parameters $\alpha$ and $\beta$ in Procedure 3 are the start window and the forward window size respectively. The detected onset positions are forwarded by a forward window, in case the actual onsets do not appear at the beginning of the transient. Afterwards, it is assumed the start of the transient is the minimum point of the first 10%

of the "note". In this section, by "note", the author refers to the interval between two onsets. Since the roughness can affect the performance, a smoothing filter is applied to the RMS curve before the process. A Kolmogorov - Zurbenko (KZ) filter [Yang and Zurbenko, 2010] with 10 samples window size and 5 iterations is used. Compared to a moving average filter, this type of filter has a better performance of attenuating the frequency components above the cutoff frequency [Yang and Zurbenko, 2010].

**Procedure 3** Calculate features designed for mono loops.

**Input:**
$\mathcal{A}$ = Audio_Loop ; $\alpha = 10\%$; $\beta = 0.2$ms.
**Output:**
$T_A$; $A1_{att}$; $A2_{att}$.

$K$ = OnsetEventDetection($\mathcal{A}$)
$k \in K$, $K$ = *all the onset positions in the given loop*
**for** $k_i \in K$ **do**
  **if** $k_i - k_{i-1} < 1ms$ **then**
    skip;
  **end if**
  $k_i = k_i - \beta$
  $R$ = RMS($\mathcal{A}[k_{i-1} : k_i]$)
  $C$ = KZ_filter($R$)
  $s$ = argmin($C[0 : \alpha]$)
  $p$ = argmax($C$)
  $e$ = size($C$)
  $[a1, b1, c1]$=fit_poly($C[s : p]$)
  $[a2, b2, c2]$=fit_exp($C[p : e]$)
  **if** $(a1 > 0 \wedge -b1/2a1 > \beta) \vee (a1 < 0 \wedge -b1/2a1 < \beta) \vee (a_2/(e - p - b_2) > 0)$ **then**
    skip;
  **end if**
  $t_i = T_A(C)$; $a1_i = A1_{att}(C)$; $a2_i = A2_{att}(C)$
**end for**
$T_A = average(t)$; $A1_{att} = average(a1)$; $A2_{att} = average(a2)$

### 4.3.2 NMF

The second decomposition method uses spectral modelling, i.e., Non-negative matrix factorisation. NMF (c.f. Equation. 4.9) aims at decomposing the matrix **V** into a product of two non-negative matrices **W** and **H**. The target matrix **V** is the magnitude spectrogram of a the audio. In the case of this Thesis, the objective is to decompose a loop with potentially overlapping sound events into individual events so that compression related features can be more readily extracted. The spectrogram is calculated using a window size of 4096 samples and an overlap of 1024 samples. The matrix **W** is the dictionary which contains $C$ basis vectors. Details of obtaining the dictionary will be given in the following paragraphs. Meanwhile the matrix **H** is the activation pattern corresponding to each basis vector. In

this approach, the author uses the activations $\mathbf{H}$ instead of the actual audio waveform to extract features. Since each row of $\mathbf{H}$ corresponds to a specific fraction of the loop, it is sparse and hence it can be used to retrieve the attack/release phases.

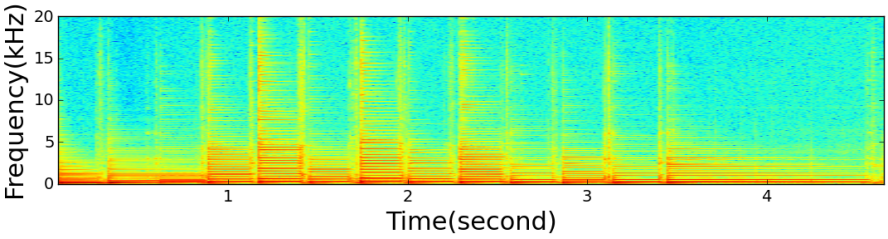
$$\mathbf{V}^{M\times N} \approx \mathbf{W}^{M\times C} * \mathbf{H}^{C\times N} \tag{4.9}$$

Unsupervised NMF suffers from a common limitation related to the dictionary recovery problem [Wang, 2017]. In another word, the objective function of NMF is to minimise the distance between the input spectrogram and the one reconstructed by multiplying the spectral template matrix and the activation matrix. It may not produce a meaningful decomposition that would serve our purpose. Reasonable results can only be obtained for simple loops with only a small amount of non-overlapping notes. Without prior knowledge on the basis vectors, the activations may not correspond to note events the author wishes to characterise. Informal observations confirmed these limitations.

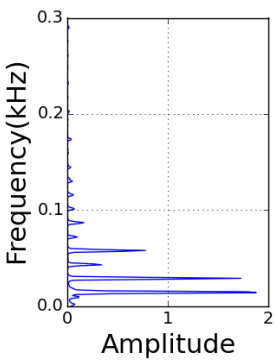
To reduce the influence of this problem, semi-supervised NMF has been used. In a real world scenario, given a random loop, pre-trained dictionary based on the notes within this specific loop is not available. Therefore, the author proposes an alternative instrument specific method. Recent works on NMF based audio information retrieval methods are built upon fixed spectral templates representing harmonic components [Bertin et al., 2010] or trained in an instrument specific manner [Benetos et al., 2014]. Similarly, this research uses a set of twelve tone equal temperament acoustic guitar notes from RWC [Goto et al., 2003] library as the template set. This solution made the pre-trained dictionary sensitive to acoustic guitar timbre as well as the instrument's pitch range. Forty-eight guitar notes across 3 octaves are used to form 4 such sets as training data for our dictionary, i.e. $w_i \in [w_1, w_2, ..., w_C]$, $w_i = 1/4 \sum_{j=1}^{j=4} w_{ij}$, with $C = 12$.

This dictionary is tested and verified on different acoustic guitar loops from the AppleLoops[1] library. An example of a loop which contains 13 notes is displayed. Its magnitude spectrogram is given in Figure 4.2(a). One dictionary element $\mathbf{w_{12}}$ from the fixed matrix $\mathbf{W}$ is given in Figure 4.2(b) which corresponds to the first activation pattern from the top in Figure 4.2(c). Although it is not possible to deliver perfect decomposition, it shows significant improvement over unsupervised NMF. Similar results are observed for other

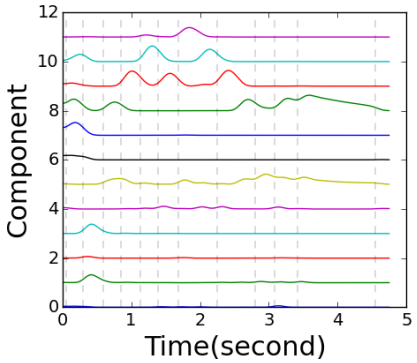
[1] `https://support.apple.com/kb/PH13426?locale=en_US&viewlocale=en_US`




(a) Spectrogram




(b) Template No.12




(c) Activation pattern

Figure 4.2: The spectrogram of an acoustic guitar loop (a), one of the fixed note template (b) and its decomposed activation pattern (c), using semi-supervised NMF.

acoustic guitar loops.

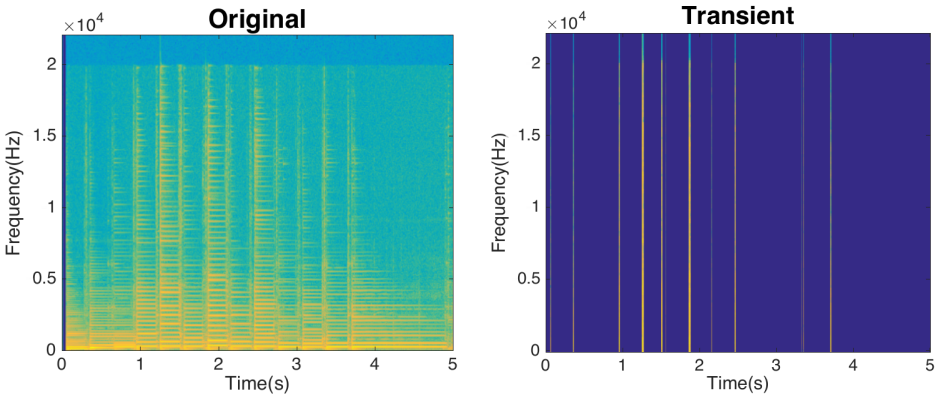
The activation curves are examined to see if they are similar to the actual energy curves when compressing the audio. The test shows positive results, since the activation curves are essentially the responses of individual notes. The activation patterns have a clear note-like shape and are sparse in general. As a result, there is no need to apply the complex selection strategies in Procedure 3, which makes this a more stable solution. The author then calculates and average using Equation. 4.6-4.8 from each activation and use the results as features.

### 4.3.3 Transient/Stationary audio separation

The final approach proposed in this part of the study is the decomposition of loops into transient and stationary ($T/S$) parts instead of individual notes. A state-of-the-art algorithm is proposed in [Siedenburg and Dörfler, 2013] using Iterated Shrinkage/Threshold Algorithm [Beck and Teboulle, 2009] framework, with a Matlab toolbox implementation[2]. An improvement over this using cross shrinking is proposed in [Siedenburg and Doclo, 2017] which provides good results for this case. This algorithm has been introduced in the

[2] `https://kaisiedenburg.net/research/`

Background Chapter, c.f. Section 2.3.3. An example of the separation is shown in Figure 4.3(a), 4.3(b). This algorithm is able to retrieve the start and stop positions of both transients and the stationary parts. The retrieved transient positions can be used as $N_{startA}$ and $N_{endA}$ in Equation. 4.6-4.8 for attack features, and the stationary positions can be used for release features. The author then computes features similarly to the previous cases.




(a) Spectrogram (b) Transient

Figure 4.3: The spectrogram of an acoustic guitar loop and its transient positions, using ISTA.

## 4.4 Feature Selection

In the previous sections, a full feature set for isolated notes are presented in Section 4.2. The handcrafted features are further developed in Section 4.3 for more complex audio materials. In this section, a series of feature selection strategies are tested aiming at achieving better performance as well as removing redundancy.

The two typical categories of feature selection are model dependent methods and model independent methods. Others may refer to them as "wrapper model" and "filter model", c.f. Section 2.5.2. The wrapper models rank features by measuring system performance. This can be fitted into any machine learning framework, however, it may suffer from overfitting [Bennasar et al., 2015]. The filter methods rank features by measuring relevance between the feature and a label or across features. The measurement may be correlation or mutual information. Filter models normally have less computational cost compared to wrapper models, however, they do not consider performance. This drawback may lead to suboptimal selection of features from the perspective of the learning task. The details of

the models are reported along with the selection results of one example, the 24 features for attack time, in Section 4.4.1, 4.4.2, 4.4.3. The audio set is generated by manually compressing 60 violin notes from RWC isolated note database [Goto et al., 2003] using 100 attack time settings within the range of (0,100]ms using the DRC designed in SAFE project [Stables et al., 2014]. The features are extracted from the audio above. Section 4.4.3 illustrates two strategies this research employed specifically for random forest regression. The same procedures are applied to the rest of the three parameters and the final results are presented in Section 4.5.3.

To simmarise and recap, Table 4.1 is provided here that sort abbreviations of each audio feature into categories. To be concise, only the categories are displayed in the table. Details of each feature can be found in Section 4.2.

| Domains | Types | Feature abbreviations |
|---|---|---|
| Frequency domain | Spectrogram based | $SC_{mean}$, $SC_{var}$, $SV_{mean}$, $SV_{var}$, $SS_{mean}$, $SS_{var}$, $SK_{mean}$, $SK_{var}$ |
| | Melspectrogram based | $MC_{mean}$, $MC_{var}$, $MV_{mean}$, $MV_{var}$ |
| Time domain | Sample based | $T1_{mean}$, $T1_{var}$, $T2_{mean}$, $T2_{var}$ |
| | RMS based | $RMS_{mean}$, $RMS_{var}$ |
| Designed feature | Ratio feature | $R_a$ |
| | Attack time feature | $T_A$, $A1_{att}$, $A2_{att}$, $A3_{att}$, $A4_{att}$, $A5_{att}$ |
| | Release time feature | $T_R$, $R1_{att}$, $R2_{att}$, $R3_{att}$, $R4_{att}$, $R5_{att}$ |

Table 4.1: Summary of the feature abbreviations.

### 4.4.1 Filter Model

*I. Ranking features based on the relevance between the features and the label*

The first and simplest strategy of feature ranking using the filter model is based on the relevance between the label and the feature, where the labels are the parameter values used as training target for the regression model. The Pearson correlation coefficient [Benesty et al., 2009] and the adjusted mutual information [Vinh et al., 2010] are calculated as the two measurements for relevance. This strategy assumes that the higher the correlation or mutual information is, the more important the feature is. The ranking results are given in Table 4.2 for the attack time feature set. Different superscript are used to represent three types of features. $^\star$ and blue text is for frequency domain features, $^\dagger$ and red text for handcrafted features, and the rest are temporal features.

| | |
|---|---|
| Corr | $RMS_{mean}$ >$T1_{mean}$ >$MV^{\star}_{var}$> $T2_{mean}$ > $MC^{\star}_{var}$> $RMS_{var}$ >$T1_{var}$ >$A3^{\dagger}_{att}$>$T2_{var}$ > $A4^{\dagger}_{att}$> $A5^{\dagger}_{att}$> $A2^{\dagger}_{att}$> $A1^{\dagger}_{att}$> $T^{\dagger}_{A}$>$SK^{\star}_{var}$> $SC^{\star}_{var}$>$SK^{\star}_{mean}$> $SK^{\star}_{mean}$>$SV^{\star}_{var}$> $SS^{\star}_{mean}$ >$SV^{\star}_{mean}$>$SC^{\star}_{mean}$>$MV^{\star}_{mean}$>$MC^{\star}_{mean}$ |
| Mu_info | $MV^{\star}_{mean}$>$MV^{\star}_{var}$>$MC^{\star}_{mean}$> $RMS_{mean}$ > $T1_{mean}$ > $T2_{mean}$ > $MC^{\star}_{var}$> $T1_{var}$ > $RMS_{var}$>$A3^{\dagger}_{att}$> $T2_{var}$ > $A2^{\dagger}_{att}$> $A4^{\dagger}_{att}$> $A5^{\dagger}_{att}$ >$A1^{\dagger}_{att}$>$T^{\dagger}_{A}$> $SC^{\star}_{mean}$> $SV^{\star}_{mean}$> $SC^{\star}_{var}$ > $SV^{\star}_{var}$>$SS^{\star}_{mean}$>$SK^{\star}_{mean}$> $SK^{\star}_{var}$> $SV^{\star}_{var}$ |

Table 4.2: Ranking for attack time features based on two relevant measure, *Corr* for cross-correlation, and *Mu_ info* for mutual information.

Both methods tend to choose temporal features in a higher ranking position than frequency domain features, except for the MFCC features in the mutual information case. It partially proves the assumption that the frequency domain features cannot provide sufficient information for attack and release time prediction. This theory is investigated further in subsequent feature selection experiments using different methods.

*II. Ranking features based on the relevance across the features*

The previous ranking method is able to show the relevance between features and the label. However, it does not exploit redundant information between features or discard features that contain overlapping information. It is possible that two features are highly related and actually using only one is sufficient. For this reason the relevance across all features is examined.

Figure 4.4 shows a dendrogram resulting from clustering features using mutual information. For the purpose of demonstration, the figure plots $1 - mutual\_info$. The result seems reasonable since it groups temporal features together. The same effect is observed for frequency domain features as well. The rule here is to choose the features such that redundant information is reduced. If two features have a high mutual information, this strategy would use only one of them instead of both. Based on this rule, a threshold is set and all features within the clusters which have the mutual information above the threshold are selected. For the clusters lower than the threshold, one feature will be selected using the Max-Relevance and Min-Redundancy (mRMR) strategy [Peng et al., 2005]. The condition is described in Equation. 4.10, where $X$ represents the full feature set, and $S_m$ is the $m$-sized cluster where one feature needs to be selected. The condition maximises the mutual information between the feature and label while minimises the mutual information

for the selected feature and all the features outside of the selected cluster. In this test, the threshold is experimentally set to 0.5.

$$\max_{x_j \in S_m} [I(x_j; c) - \frac{1}{m} \sum_{x_i \in X - S_m} I(x_j; x_i)] \tag{4.10}$$

where $X$ represents the full feature set, and $x_i$ is one of the feature. $c$ represents the label, and $I$ is the operation of calculating the mutial information.

The resulting selected features are as follows:

$MC_{mean}, SC_{var}, SK_{mean}, SV_{mean}, T_A, A5_{att}, A3_{att}, MC_{var}$;

Since this method compares relevance across features, the clustering tends to put the same type of features together, e.g. frequency domain features are grouped together. Theoretically, the repeated features are discarded in terms of mutual information. However, the same as all other filter models for feature selection, this process does not consider if the features are related to the problem. This method will yield features that provide the most information, but not necessarily the ones most related to the target label.

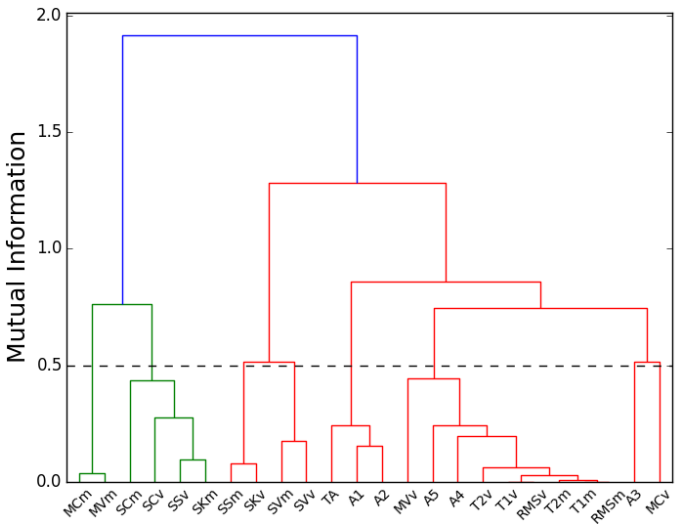



Figure 4.4: Relevance between features

### 4.4.2 Wrapper model

In the next stage, the author applies the wrapper model of feature selection suggested in [Baume et al., 2014], and originally introduced in [Liu and Yu, 2005]. The selection strategy

is described using the pseudo-code in Procedure 4. This algorithm avoids exhaustive search and hence reduces the computation time significantly. $N$ in the pseudo-code represents the feature set size set to 24. At the starting point of the algorithm $n$ is set to 2. For each iteration, top $m$ feature sets are passed to the next step, this experiment sets $m = 6$. The algorithm stops when the sub_set features size equals to the full_feature size. The best_features are sorted according to the regression performance. Repeated random sub-sampling validation (Monte Carlo variation [Burman, 1989]) is used for evaluation, such that the dataset is split into 90% training and 10% testing. The process is repeated 100 times and the average of the Mean Absolute Error (MAE) is used as the performance measure. The best prediction accuracy is displayed in Figure 4.5. The best performance is provided when using 7-10 features, since the value is very close in this range. It is reasonable to choose fewer features in wrapper models to reduce overfitting. In this case, the author chooses eight features:

$$T_A, A1_{att}, T1_{mean}, T2_{mean}, MC_{mean}, MC_{var}, MV_{mean}, MV_{var}.$$

The wrapper model is able to provide the best possible accuracy, but it may be overfitted to this particular dataset and therefore may lose its universality or generalisability.

**Procedure 4** Feature selection using wrapper model

```
full_set = D(F_0, F_1, ..., F_{N-1})
sub_set = combination(N, n)
for i ∈ [n + 1 : N] do
  best_feature = sort(evaluation(∀{sub_set}))
  sub_set = best_feature[0:m]
  for j ∈ [1 : m] do
    for k ∈ [1 : N] do
      if full_set[k] ∉ sub_set[j] then
        sub_set[j].append(sub_set[j], full_set[k])
      end if
    end for
  end for
end for
```

### 4.4.3 Feature significance

In this research, the author also considers two methods specifically designed for the random forest algorithm. The first method randomises the value of a certain feature and use the change in the out-of-bag ($OOB$) error rate to assess feature significance. Assuming there is a feature set $\mathbf{X} = \{X_0, ...X_j, ...X_M\}$, and the task is to rank the $M$ features. The algorithm grows $T$ decision trees. For each decision tree $t$, the prediction error is calculated using the

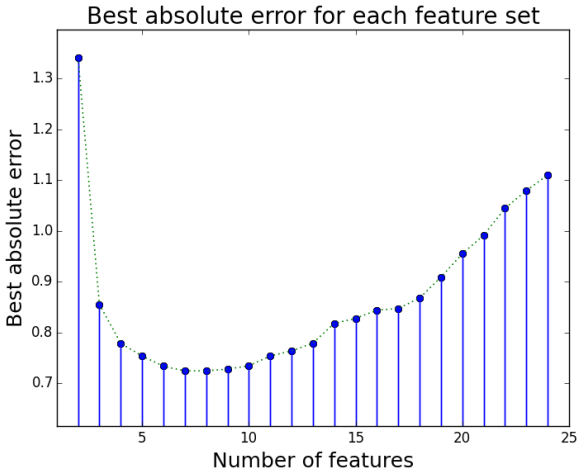



Figure 4.5: Accuracy performance for the wrapper model with the increase of the number of features used.

out-of-bag samples $OOB_t$ as $\epsilon = errOOB_t$. If replacing $X_j$ to random values, it will lead to a new error $\widetilde{\epsilon}_t$. The variable importance is defined as $VI(X_j) = 1/T \sum_t^T (\widetilde{\epsilon}_t - \epsilon_t)$. In this implementation, the feature set size $M$ is set to 24, and the amount of decision trees in the forest is set to $T = 10$. The top 10 most significant features are selected as follows:

$$MC_{mean}, MC_{var}, MV_{mean}, MV_{var}, RMS_{var}, A4_{att}, SV_{mean}, RMS_{mean}, A5_{att}, A2_{att};$$

The second approach uses the decrease in node impurity to decide on the feature importance. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two, so that similar response values end up in the same set. The optimal condition is chosen based on variance in the case of regression, and this measure is called impurity. When training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure. For this method, the amount of trees $T$ need to be larger than the number of features $M$. Therefore the author chooses $T = 100$. The most important features are chosen as follows:

$$A3_{att}, MC_{mean}, MC_{var}, SC_{var}, SS_{mean}, A2_{att}, A5_{att}, A1_{att}, MV_{var}, MV_{mean};$$

## 4.5 Evaluation

To recap, the workflows in Section 4.2, 4.3, and 4.4 firstly provides a full feature set. Then the handcrafted features are developed targeting more complex audio materials, mono-instrument loops. A series of feature selection strategies are tested to choose the optimal

feature set for each DRC parameter.

In this section, two stages of evaluation will be performed. The first stage evaluates the features' efficiency when developed in the context of mono-instrument loops. The process starts from a numerical test presented in Section 4.5.1. It evaluates the regression model trained by the handcrafted features designed in Section 4.3. The predicted mean absolute error (MAE) of each parameter is reported when using each decomposition method to extract features. The other evaluation test is a similarity test using an audio similarity model discussed in Section 4.5.2. The audio materials for this experiment are 29 acoustic guitar, 30 electronic bass, and 12 drum loops from AppleLoop. The second stage evaluates the feature selection strategies and provides the final selection results for all DRC parameters. The evaluation across each selection algorithms will be presented, along with the relations across each parameter. The feature set is generated using 60 violin notes from the RWC isolated note database. Feature sets for threshold, ratio, release time are calculated in the same fashion as the attack time test outlined in Section 4.4, where the author manually sets 100 settings for each parameter within (0,50]dB for threshold, [1,20] for ratio, (0,1000]ms for release time using the SAFE DRC. For the random forest regression model, the feature sets are the training data while the training targets are the parameter values.

### 4.5.1 Numerical accuracy test for DRC specific features

Since the overall numerical evaluation for all four parameters has been done in Chapter 3, and the results show it is harder to predict the ballistic parameters, the handcrafted features proposed in this section focus on attack time and release time. The dataset is generated by compressing $N$ loops respectively to both parameters and leave the other parameters as default, (0,100]ms for attack time with step of 1ms, and (0,1000]ms for release time with step of 10ms. Therefore, there are $N * 100$ compressed audio excerpts respectively. The model is aimed at learning the difference between the *Input* and the *Reference*, c.f. Figure 3.1. The training data is formed by extracting features from each compressed audio and dividing them by the features extracted from the originals. The corresponding compressor parameters are used as training target for a random forest regression model. The validation method in this experiment is random sub-sampling validation (Monte Carlo variation). Twenty percent of each feature vectors are selected for testing, while the remaining data are used for training. The process is repeated 100 times and the average MAE is reported in Table 4.3. $\tau_a$ stands for attack time and $\tau_t$ for release time. *Std* stands for standard features,

which represents the 6 high order statistical features c.f. Section 4.2. The following labels, *Onset*, *NMF*, and *T/S*, represent the feature sets that contain both standard features and the ones extracted using the labelled note event detection method.

For the test cases, the error drops when the standard features and the DRC specific features extracted using the decomposition methods are combined. NMF features provide the best performance compared to the other individual features. However, using all features together produces the lowest error rate. Therefore, even though NMF stands out in this numerical evaluation, instead of choosing this specific feature, it is possible to use all three together for a better performance.

| MAE(ms) | *Std* | *Onset* | *NMF* | *T/S* | *All* |
|---|---|---|---|---|---|
| $Guitar, \tau_a$ | 0.934 | 0.897 | **0.845** | 0.863 | **0.807** |
| $Bass, \tau_a$ | 1.449 | 1.196 | **1.071** | 1.244 | **0.995** |
| $Drum, \tau_a$ | 1.384 | 1.361 | **1.194** | 1.274 | **1.134** |
| $Guitar, \tau_r$ | 12.115 | 10.604 | **10.442** | 11.802 | **9.981** |
| $Bass, \tau_r$ | 11.701 | 11.143 | **10.733** | 10.886 | **9.381** |
| $Drum, \tau_r$ | 16.327 | 14.946 | **12.714** | 13.315 | **12.043** |

Table 4.3: Predicted Mean Absolute Error(MAE) using different feature sets for loops of three instruments.

### 4.5.2 Similarity test for designed features

In the previous section, the dataset is split into training and testing set to evaluate the efficiency of the prediction model. In a more realistic situation, the *Reference* and the *Input* should be independent (for more details see Figure 3.1 and the similarity simulation in Section 3.2.2). In this section, 50 pairs of audio are randomly selected in the audio set, using one as reference and the other one as input audio. The model is able to give a predicted parameter set according to these two inputs. An output audio can be generated accordingly. Therefore, the system can be evaluated by comparing $D1$ and $D2$, which represented the dissimilarity between input and reference, and prediction and reference respectively (c.f. Equation.4.11). $D()$ represents the dissimilarity measure function. Theoretically, the distance between prediction and reference should be smaller than the distance between input and reference.

$$
\begin{aligned}
D1 &= D(Output, Reference); \\
D2 &= D(Input, Reference);
\end{aligned}
\tag{4.11}
$$

A simple audio similarity model is used to test the efficiency of the system, which is also used in the author's earlier research [Sheng and Fazekas, 2017] and Chapter 3 . Details demonstrated in Section 3.2.2. A further development on the similarity measurement will be delivered in Chapter 6. To recap, the model extracts MFCC coefficients and they are used to fit a Gaussian Mixture Model(GMM). An approximation of the symmetrised KL divergence is then calculated and used as a dis-similarity measure. The average of 50 cases are displayed in Table 4.4. Results show $D2$ are smaller than $D1$ for all cases, which means this method is able to bring the *Output* close to the *Reference* compared to the *Input*. Since the actual value of the divergence does not have practical meaning, $D2$ is normalised according to $D1$, i.e. set $D1 = 1$, and only the normalised results are reported.

| | $D2_{std}$ | $D2_{onset}$ | $D2_{nmf}$ | $D2_{t/s}$ | $D2_{all}$ |
|---|---|---|---|---|---|
| $Guitar, \tau_a$ | 0.918 | 0.916 | **0.914** | 0.916 | **0.916** |
| $Bass, \tau_a$ | 0.384 | 0.375 | **0.371** | 0.383 | **0.362** |
| $Drum, \tau_a$ | **0.251** | 0.252 | **0.251** | 0.257 | 0.252 |
| $Guitar, \tau_r$ | 0.934 | 0.936 | 0.940 | **0.919** | **0.917** |
| $Bass, \tau_r$ | 0.738 | 0.732 | **0.726** | 0.733 | **0.729** |
| $Drum, \tau_r$ | 0.583 | 0.589 | **0.580** | **0.582** | 0.584 |

Table 4.4: $D1$ and $D2$ comparison using different feature sets, when $D()$ is the audio perceptual similarity.

The trend from the numerical test is not fully consistent with the similarity test. The top two closest distances in each case are highlighted. NMF still outperformed the other decomposition methods, however, the closest distance does not always appear when using all three types decomposition methods. One observation is that the average similarities are not distinguishable between different feature sets. The author then examined the individual predictions. The predicted parameter values are rather close (<1ms, c.f. Table 4.3), correspondingly the outputs of the similarity model are very close. It is reasonable because even if the features are extracted by different decomposition methods, it is the exact same features that are extracted after decomposition. They are designed to provide similar information. The difference is their efficiency and complexity. Considering the results from

both evaluation processes, it can be stated that the most efficient decomposition method is NMF both numerically and perceptually, while using all three sets of features together, the proposed model is able to provide better numerical performance.

### 4.5.3 Overall performance of feature selection

In this and the subsequent section, the evaluation of the feature selection algorithms will be discussed. Six selection algorithms are demonstrated in Section 4.4, where the features for attack time are used as an example. Based on the selection results from six algorithms, this research consider the wrapper model in advance of other models, since it will guarantee optimal performance. The balancing strategy is to choose the wrapper model results plus the features that are selected more than four times among the five algorithms. The selected feature set is given in Table 4.5 for all four parameters.

| Parameters | Selected Features |
|---|---|
| Threshold | $MC_{mean}$, $MC_{var}$, $MV_{mean}$, $RMC_{mean}$, $SC_{mean}$, $SC_{var}$, $SV_{mean}$, $SV_{var}$, $SS_{var}$, $SK_{mean}$, $SK_{var}$. |
| Ratio | $MC_{var}$, $MV_{var}$, $T1_{mean}$, $RMC_{mean}$, $RMC_{var}$, $SC_{mean}$, $SC_{var}$, $SV_{mean}$, $SV_{var}$, $SS_{mean}$, $SS_{var}$, $SK_{var}$, $R_a$. |
| Attack time | $T1_{var}$, $T2_{var}$, $A3_{att}$, $A1_{att}$, $T_A$, $A2_{att}$, $A5_{att}$, $MC_{mean}$, $MC_{var}$, $MV_{mean}$, $MV_{var}$. |
| Release time | $T_R$, $A1_{rel}$, $A2_{rel}$, $A4_{rel}$, $A5_{rel}$, $T1_{mean}$, $T1_{var}$, $RMS_{mean}$, $RMS_{var}$, $MC_{mean}$, $MV_{var}$, $SV_{mean}$. |

Table 4.5: The final selected features for four parameters, balancing the selection models.

| Parameters | Threshold | Ratio | Attack | Release |
|---|---|---|---|---|
| *Selected* | **1.242dB** | **0.919** | **0.830ms** | **9.265ms** |
| *Full-list* | 1.295dB | 0.950 | 1.122ms | 12.572ms |
| *Corr* | 1.808dB | 1.103 | 0.978ms | 12.259ms |
| *Mu_info* | 1.461dB | 0.934 | 0.837ms | 12.157ms |
| *Across* | 1.663dB | 1.016 | 1.147ms | 11.635ms |
| *RF_1* | 1.452dB | 0.987 | 0.908ms | 12.604ms |
| *RF_2* | 1.580dB | 1.098 | 0.982ms | 13.808ms |
| *Wrapper* | **1.218dB** | **0.892** | **0.725ms** | **8.759ms** |

Table 4.6: Prediction MAE comparing the selected features, full set, and individual selection results.

The selection results for threshold and ratio show that the most related features for

these parameters are frequency domain features and MFCC features, while for attack time and release time, the frequency domain features are the least frequently selected. MFCC features are the most popular among all four features, due to their relation with frequency envelope as well as timbre. The prediction error comparison across models are provided in Table 4.6, where the values are the average of MAE calculated through Monte Carlo variation. Here, 20% of the dataset are randomly split as testing data 100 times and the average MAE is reported. Variances across each random validation are very small, about 0.006 for threshold, 0.01 for ratio, 0.007 for attack time and 0.7 for release time in average indicating a stable performance.

The final selected feature sets balanced all selection results. The performances are comparable with the best performance selected by the wrapper model, and better than the filter models, random forest feature importance methods, and the full feature set. The results indicate that the selection improves the error rate, and the selected feature sets are much smaller in size than the full feature set, which also reduces the computational cost.

The selection results of each algorithm and each parameter are represented in Figure 4.6-4.9. The results for threshold and ratio show a preference for frequency domain statistical features. The most commonly selected features in case of threshold are $SC_{mean}, SV_{mean}$, $SK_{mean}, MC_{mean}$, and $SS_{mean}$ for ratio. The most commonly selected feature for attack time is $MC_{var}$ which has been selected by all methods. For release time, it is $MV_{var}$ which is an MFCC derived feature as well. The features designed specifically for the attack/release phase are also selected frequently for these two parameters. Figure 4.8 shows a clear trend that all methods overlook frequency domain features, which fits the assumption that conventional features from literature are not the best choices when predicting these parameters. Figure 4.9 does not show exactly the same trend as Figure 4.8, however, the wrapper model does not choose any frequency domain feature, which means even with a certain relevance, frequency domain features may harm the performance (c.f. Figure 4.5, after the optimised feature set, adding more features increases the error rate). Therefore, it can be stated that conventional features are satisfactory to predict threshold and ratio, but to predict attack time and release time, the specifically designed features are necessary.

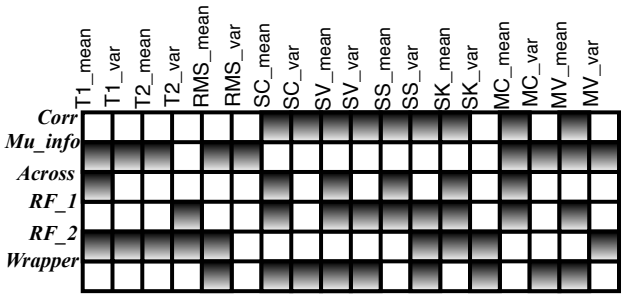



Figure 4.6: Selection results of 6 algorithms for threshold. Grey blocks represents the features that are selected by this method.

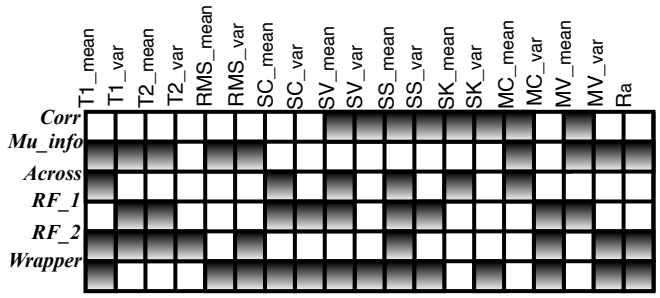



Figure 4.7: Selection results of 6 algorithms for ratio. Grey blocks represents the features that are selected by this method.

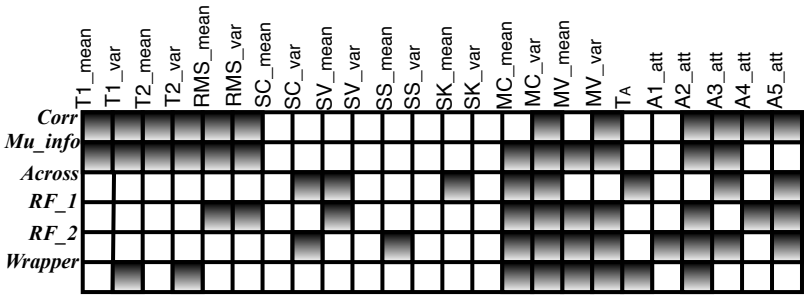



Figure 4.8: Selection results of 6 algorithms for attack time. Grey blocks represents the features that are selected by this method.

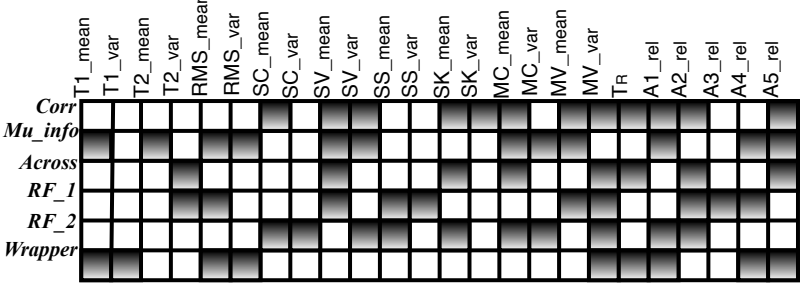



Figure 4.9: Selection results of 6 algorithms for release time. Grey blocks represents the features that are selected by this method.

### 4.5.4 Relations across parameters

The observation in the overall performance is that the feature selection methods for attack and release time tend to select similar features, while threshold and ratio likewise exhibit similar behaviour. The overlap rate of the selected features across two pairs and six algorithms are displayed in Table 4.7. Number $I$ to $VI$ represent $Corr, Mu\_info, Across, RF\_1$, $RF\_2$, and $Wrapper$ as in Table 4.6 respectively. Except for the correlation selection result for attack and release time, all overlap rates are higher than 50%. The results fit the assumption that threshold and ratio have their similarity since they are more directly affecting dynamic range. Attack time and release time are also similar due to the fact that they are both timbre related parameters and they both related to the speed of the DRC's action.

| Overlap | *I* | *II* | *III* | *IV* | *V* | *VI* |
|---|---|---|---|---|---|---|
| Attack/Release | 0.17 | 0.58 | 0.50 | 0.60 | 0.70 | 0.56 |
| Threshold/Ratio | 0.89 | 0.89 | 1.00 | 0.56 | 0.56 | 0.69 |

Table 4.7: Feature overlap rate between parameter pairs

### 4.5.5 Relations across selection algorithms

In this section, the overlap rate of the selected features across each selection algorithm is analysed for each parameter. Since different algorithms do not guarantee the selection of the same amount of features, the overlap tables do not represent diagonal matrices. The overlap rate for row $i$ and column $j$ is calculated as follows: $rate = \#overlap(i,j)/\#feature(i)$. Table 4.8-4.11 represent the overlap rates for four parameter features.

| Overlap | *I* | *II* | *III* | *IV* | *V* | *VI* |
|---|---|---|---|---|---|---|
| *I* | 1 | 0.22 | 0.56 | **0.89** | 0.22 | 0.67 |
| *II* | 0.22 | 1 | 0.22 | 0.22 | **0.56** | 0.33 |
| *III* | **0.83** | 0.33 | 1 | **0.83** | 0.17 | 0.33 |
| *IV* | **0.89** | 0.22 | 0.56 | 1 | 0.33 | 0.56 |
| *V* | 0.22 | **0.56** | 0.11 | 0.33 | 1 | 0.33 |
| *VI* | **0.67** | 0.33 | 0.22 | 0.56 | 0.33 | 1 |

Table 4.8: Feature overlap rate across 6 algorithms for threshold

Comparing Table 4.8-4.11, one of the common trend is that the wrapper model, as *VI*, has the highest overlap with the filter models using correlation and mutual information. It indicates that the features that are able to produce the optimal result are the ones that have

| Overlap | *I* | *II* | *III* | *IV* | *V* | *VI* |
|---|---|---|---|---|---|---|
| *I* | 1 | 0.22 | 0.56 | 0.56 | 0.11 | **0.67** |
| *II* | 0.22 | 1 | 0.33 | 0.33 | **0.67** | 0.56 |
| *III* | **0.83** | 0.50 | 1 | 0.50 | 0.33 | 0.50 |
| *IV* | 0.56 | 0.33 | 0.33 | 1 | 0.44 | **0.67** |
| *V* | 0.11 | **0.67** | 0.22 | 0.44 | 1 | **0.67** |
| *VI* | **0.46** | 0.38 | 0.23 | **0.46** | **0.46** | 1 |

Table 4.9: Feature overlap rate across 6 algorithms for ratio features

| Overlap | *I* | *II* | *III* | *IV* | *V* | *VI* |
|---|---|---|---|---|---|---|
| *I* | 1 | **0.75** | 0.25 | 0.58 | 0.42 | 0.42 |
| *II* | **0.75** | 1 | 0.25 | 0.58 | 0.50 | 0.58 |
| *III* | 0.38 | 0.38 | 1 | 0.50 | **0.63** | 0.38 |
| *IV* | **0.70** | **0.70** | 0.40 | 1 | 0.60 | 0.50 |
| *V* | 0.50 | **0.60** | 0.50 | **0.60** | 1 | 0.50 |
| *VI* | 0.56 | **0.78** | 0.33 | 0.56 | 0.56 | 1 |

Table 4.10: Feature overlap rate across 6 algorithms for attack time features

the strongest relevance with the label. However, the two types of filter models do not have a high overlap rate, which suggests correlation and mutual information do not necessary select the same features, which is well known from a theoretical perspective as discussed in [Li, 1990]. The results from this research corroborate this theory and also suggest that it is reasonable to run both strategies and balance the results. The model comparing relevance across features, *Across*, as *III*, in the tables, shares the lowest overlap rate with the other methods. This method guarantees the least mutual information across the selected features, but it does not consider any relation between the features and the label. This is the major difference between this and all the other selection methods. Conversely, the filter model using mutual information, *II*, as $Mu\_info$, in the tables, shares the most overlap with other methods, which shows it is an efficient method on its own.

## 4.6 Conclusion

This chapter started with a description of a thorough feature set for predicting four of the important DRC parameters. The handcrafted features are extended by decomposing audio loops and extracting note structure related information. The use of these features have shown to be beneficial for the proposed framework for the intelligent control of the

| Overlap | *I* | *II* | *III* | *IV* | *V* | *VI* |
|---|---|---|---|---|---|---|
| *I* | 1 | 0.25 | **0.58** | 0.33 | **0.58** | 0.33 |
| *II* | 0.25 | 1 | 0.25 | 0.33 | 0.33 | **0.50** |
| *III* | **0.88** | 0.38 | 1 | 0.63 | 0.50 | 0.38 |
| *IV* | 0.40 | 0.40 | **0.50** | 1 | 0.30 | 0.40 |
| *V* | **0.70** | 0.40 | 0.40 | 0.30 | 1 | 0.20 |
| *VI* | 0.40 | **0.60** | 0.30 | 0.40 | 0.20 | 1 |

Table 4.11: Feature overlap rate across 6 algorithms for release time features

dynamic range compressor in this Thesis. The benefit is clear both in terms of the accuracy of predicting attack and release times as well as audio similarity using a simple perceptual model. Overall results show that using all three feature sets works best numerically, while NMF stands out in both numerical and perceptual tests.

In this chapter, the author also introduced a feature selection experiment using six different selection strategies. The final selection is detailed in Section 4.5.3. The results fit the assumption that using a smaller set of features is beneficial to reduce noise, computational time and improve the performance at the same time. The results also show that frequency domain features are less efficient when predicting attack and release time, while the opposite is true when predicting threshold and ratio. The results indicate that commonly used features are not sufficient when it comes to predicting the time constant parameters of the DRC. For all four parameters, MFCC related features are the most often selected, which is clearly due to their relations with both frequency domain information and timbre.

This chapter aimed at improving the first component of the proposed intelligent system, i.e. *Feature Extraction*. Feature selection results can be used as a guideline for the implementation as well as future research. A specific feature set can be applied when predicting each parameter. The feature extraction process can be improved further in different ways. It might be still possible to push the boundary of the handcrafted features, but this is out of the scope of this Thesis. There is another problem the author attempts to tackle, which is the problem of predicting four parameters jointly. In the next chapter, Chapter 5, a DNN feature learning model is proposed to serve as a feature learning scheme, which is an attempt to solve this problem. It will be compared with individual parameter prediction. The optimisation of the other components of the final proposed system for DRC parameter estimation will be provided in the following chapters of the Thesis.

# Chapter 5

# Siamese Model for Feature Learning

The research before this chapter has been focussing on simple audio materials, i.e. isolated notes and mono-instrument loops, and simple prediction situation, i.e. predicting a single parameter. Features were designed to predict each parameter individually. In this chapter, the research moves on to more complex audio materials and more complex prediction situation. This chapter focuses on learning a feature embedding from loops or polyphonic music excerpts that can be used to predict multiple parameters jointly. To tackle this problem, Deep Neural Networks (DNN) have been applied. Deep Learning is the most popular and revolutionary scientific research trend in recent years. Deep Learning normally refers to machine learning research using DNN. In this research, the advantages of DNN are used to enhance the feature extraction process in the system design. In the rest of this chapter, the motivation for using a DNN is provided, followed by the network designs and model parameter tuning. The evaluation and the conclusion related to this particular approach are also included to conclude the chapter.

## 5.1 Motivation

In this section, I will outline why moving beyond the previously introduced feature extraction methods may be beneficial and introduce the Deep Learning approach investigated in the rest of this chapter. Deep Neural Networks, particularly Convolutional Neural Networks (CNN) have become exceptionally successful in a wide variety of visual object

recognition and classification tasks [Krizhevsky et al., 2012]. The reasons for this are now well understood. For instance, CNNs can learn filters corresponding to increasingly complex shapes in the target image hence becoming successful at classifying or labelling images. In the domain of audio, the application of CNNs have also proved successful in several tasks including audio labelling and similarity estimation. The input representation is usually a time-frequency image, e.g. Fourier or Mel-spectrogram [Pons et al., 2017b, Choi et al., 2016, Ullrich et al., 2014], but increasingly, raw audio samples are used as well [Ardila et al., 2016, Dieleman and Schrauwen, 2014]. The reason for the success of these approaches is less straightforward to see, because there is poor analogy between shapes or objects in images and events, such as notes or chords in audio. Audio events are typically distributed in frequency, e.g. the recording of a note played on a typical instrument and its harmonic partials activate discontinuous bands along the frequency axis. The problem becomes more acute when audio events overlap.

Finding an appropriate input representation and designing a neural network suitable for recognising very specific aspects of an audio signal is also a difficult and generally unsolved challenge. Standard approaches work well for common audio classification problems, but if the task becomes focussed on a specific aspect of audio that is often obscured by large varying signal attributes, different input representations, network structures and training methods should be adopted to develop a successful solution. For instance, the dynamic range of an audio signal may be characterised by features such as the crest factor [Giannoulis et al., 2013] and also correlate with note attack and release times. These are measurable in a single note recording but become obscured by overlapping note events and other changes in complex real-world recordings.

In previous chapters, the design of an intelligent control system targeting the DRC using a reference audio is demonstrated. The key to the performance in this system is the feature extractor. Chapter 4 provides details of conventional as well as designed features used in previous research along with the optimised feature sets. Due to the fact that different feature sets are used for each DRC parameter, four individual regression models needed to be trained. A generic feature set to predict all parameters would therefore be a great benefit. In addition to the above mentioned drawbacks, most of the handcrafted features are based on note envelope structures. For audio materials like isolated notes, it is easy to extract envelopes, however for more complex audio, for instance, audio *loops* with overlapping note events, more complex algorithms like NMF and onset event detection are

required to separately analyse note events in a loop. It is still relatively straightforward to extract notes from loops, but polyphonic music brings additional difficulties to the problem and may increase the computational cost dramatically. For instance, overlapping note events may have different duration and timbre, making them difficult to identify and decompose. For the reasons above, it is reasonable to assume that a deep learning model can be beneficial in this task. Deep neural networks are capable of learning a complex nonlinear function. With an appropriate audio input representation, it should be possible to make the model learn a generic feature embedding for all four parameters. Meanwhile, having one trained model to generate features will reduce the computational complexity. The model also has the potential capability to generate efficient features regardless of audio materials. The focus of this chapter is to design a feature learning model, so that it will be possible to compare the efficiency between the trained features and handcrafted features. The features are used in conjunction with a conventional regression model, as opposed to end-to-end learning, for sake of reproducibility and easier comparison with previous approach proposed in this Thesis.

To achieve this goal, the author proposed a siamese structure with the feature embeddings formed by the difference between the output of two branches. Several architectures are tested within this two-branch framework, aiming to learn highly specialised audio features that are invariant to large variations in several other attributes. The performance is first evaluated using the baseline designs, and the model is tuned according to the task and observations. The following sections in this chapter are organised as follows: Several potentially suitable DNN model designs from the literature are adopted and evaluated in Section 5.2. The models are tuned to this specific problem by altering the model structures and hyperparameters in Section 5.3. Section 5.4 provides the evaluation results and the most suitable model tested on a larger scale and using a more complex dataset. Finally the conclusions of this chapter are outlined in Section 5.5.

## 5.2 Model Design

A neural network is a type of algorithm that uses a certain combination of artificial neurons (see Section 2.5.3 in Chapter 2 for a more in-depth introduction.) The work in this chapter has been built on one of the most popular structure, i.e. Convolutional Neural Network (CNN). This type of network is able to detect the local patterns, and the recognition can

become increasingly more complex with deeper networks.

The purpose of the work presented in this chapter is to learn a feature embedding that represents all the characteristics of the DRC. A feature embedding can be considered an intermediate output of a DNN. It is a lower dimensional vector that is learnt from the input audio. To build the training dataset, we need raw audio samples and a series of compression parameters. In this chapter, the audio materials are mono-instrument loops and polyphonic music excerpts. The details of the dataset are provided in Section 5.2.4. In this chapter, the audio data are denoted as *Input 1* (original audio), *Input 2* (reference audio) in Figure 5.1, and the ground truth parameters, i.e. training target, as *Labels*. As introduced in previous chapters (see e.g. Section 3.2.1) the four DRC parameters the research focusses on are, threshold, ratio, attack time, and release time, i.e. $\boldsymbol{\rho} = \{\theta, \gamma, \tau_a, \tau_r\}$. The main novelty introduced here is the joint estimation of these parameters.

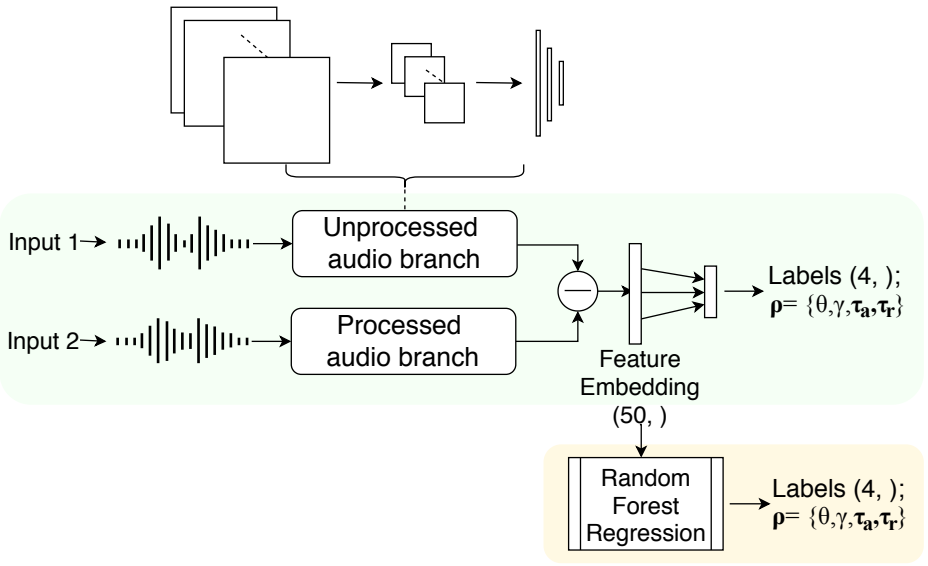



Figure 5.1: Workflow for the proposed system: it contains a twin-siamese DNN model for feature learning with the learning targets being the DRC parameters, and a random forest regressor trained using the DNN feature embedding for parameter prediction. Details of the training process is given in Section 5.2.

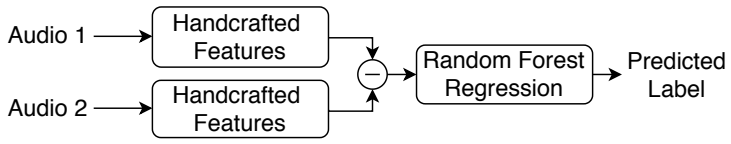
Two audio files will be used as inputs for the twin siamese model, where as in Figure 5.1, they are described as the *Unprocessed branch* and *Processed audio branch*. The identical branches contain convolutional layers as well as dense layers, and the output of them are feature embeddings. Three model designs are introduced for this branch in the following paragraph. As it is mentioned in the Background Chapter, CNN is a popular model design in both image and audio signal processing. The input representation of audio for DNN can

be a time-frequency representation or raw audio samples. There will be two model designs corresponding to these two types of input signals outlined in Section 5.2.1-5.2.2. Besides these two structures, this research also considers the multi-kernel structure to capture the features in multiple time and frequency scales. The output of the two branches, two feature embeddings, will be merged by subtraction, and this will be followed by a fully connected dense layer. The loss function the proposed method uses here is the mean squared error (MSE). The training targets are the parameters $\boldsymbol{\rho}$ of the DRC. The learning rate is updated adaptively using Adadelta [Zeiler, 2012]. The trained feature embeddings are then used as features to train a random forest regression. It follows the same procedure of the previous research, c.f. Section 3.2.2. This model is not designed as a predictor directly, but it is also possible to use a DNN regressor after the feature embedding layer. At this stage of the research, the goal is to focus on enabling the model to look for DRC related features rather than designing a well performing regressor. In the initial evaluation, it is also reasonable to use an approach that is comparable with the previous handcrafted features, and by applying random forest regression model in both cases, so the comparison can be expected to be more trustworthy.

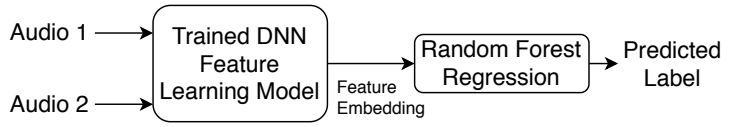
A comparison of the workflows between the approach proposed in this chapter and the previous chapters is provided in Figure 5.2. Figure 5.2(a) is reproduced from the workflow diagram in Section 3.2.2, Figure 3.3. Figure 5.2(b) is the approach the author developed in this chapter. The DNN model is pre-trained. The training dataset and process will be provided in the following sections. The feature embeddings are generated by this trained model, and then they are used to train the random forest regression model depicted in the diagrams. The workflow involves two stages of training. A more detailed workflow of this feature learning approach will be illustrated in Section 5.2.1-5.2.3.

### 5.2.1 Model design for the siamese branches - CNN structure - *Model 1*

The first model design for the identical branches is the classical CNN structure [Krizhevsky et al., 2012]. It is widely used in image processing as well as multiple audio signal processing tasks. The CNN structure has outperformed the state of the art in many research tasks, including onset event detection [Schluter and Bock, 2014], music boundary learning [Ullrich et al., 2014] and many more. The action of the DRC creates change points in the audio signal. These are not closely analogous to structural boundaries, yet it is reasonable to




(a) Handcrafted feature approach




(b) DNN learnt feature approach

Figure 5.2: Comparison of two workflows when using handcrafted and features learnt by DNN.

assume that architectures designed to detect change points might work well in our case to learn relevant features. This is reasonable, since the DRC is not operating through the whole audio, but only a certain temporal segment that is above the threshold. Within these parts, multiple aspects of the audio are changing along with different parameter settings. It can be assumed that the CNN structure is able to learn the DRC characteristics in this case.

As input representation, the commonly used Mel-spectrogram is used here (see Section 2.5.3). The first proposed network is a seven layer CNN. It consists of five convolutional layers with max-pooling, and a drop-out rate of 0.1. They are then followed by two dense layers. The model summary is provided in Table 5.1.

### 5.2.2 Model design for the siamese branches - Sample level CNN - *Model 2*

The second model structure this research proposes is to use time domain audio samples as input. As it is mentioned in the Background Chapter, a time-frequency representation is not exactly equivalent of an image signal, therefore the research also considers to use raw audio input directly. The models taking raw audio as input have previously provided equivalent performance with the ones using time-frequency representations as input, c.f. Section 2.5.3. Since DRC is operated at the audio sample level, this research applies a sample-level small filter size, and follows the model design in [Lee et al., 2017], where the proposed network contains seven 1D convolutional layers, batch-normalisation, and six layers of max-pooling. This front end is then followed by two residual layers and two

| Input: Mel-spectrogram (128,690,1) |
|---|
| Conv2D: 3*3*10<br>MaxPool2D: 2*2<br>DropOut: 0.1 |
| Conv2D: 3*3*15<br>MaxPool2D: 2*2<br>DropOut: 0.1 |
| Conv2D: 3*3*15<br>MaxPool2D: 2*2<br>DropOut: 0.1 |
| Conv2D: 3*3*20<br>MaxPool2D: 2*2<br>DropOut: 0.1 |
| Conv2D: 3*3*20<br>MaxPool2D: 2*2<br>DropOut: 0.1 |
| Flatten<br>Dense(feature embedding layer): 50<br>Dense: num_para |
| Output: Parameters |

Table 5.1: Model summary for the CNN structure, i.e. Model 1.

dense layers. The residual layers are used to avoid the vanishing gradient problem [He et al., 2016] typically caused by introducing too many layers. A summary of this model is provided in Table 5.2. Some of the convolutional layers are duplicated, the notation " *2 " is used to represent two groups of layers with the same settings. " L* " is the notation to indicate reference to specific layers.

### 5.2.3 Model design for the siamese branches - Multi-kernel CNN - *Model 3*

In this model proposal, the Multi-kernel CNN is introduced. The task is to make this siamese model learn multiple aspects of changes to sound induced by the DRC, i.e. changes in different time scales, as well as in magnitude domain, for example, attack time vs. ratio and threshold. The author considers to use the multi-kernel model construction proposed in [Pons et al., 2017b]. This model is designed to capture the audio features at multiple scales at the same time which fits the purpose of this research to observe audio characteristics over different decision horizons. Using multiple kernels in this problem is especially useful for this problem because the model needs to learn four aspects of DRC at the same time. The model with only one kernel size might neglect important audio features, such as transient features. The model applies several temporal kernel as well as timbre kernel, as it is

<table>
<tr><td colspan="3">Input: Waveform (44100,1)</td></tr>
<tr><td rowspan="5">Front End Network</td><td colspan="2">Conv1D: 3*1*64<br>Batch Normalisation</td></tr>
<tr><td>Conv1D: 3*1*64<br>Batch Normalisation<br>MaxPool1D: 3*1</td><td>*2</td></tr>
<tr><td>Conv1D: 3*1*128<br>Batch Normalisation<br>MaxPool1D: 3*1</td><td>*2</td></tr>
<tr><td>Conv1D: 3*1*256<br>Batch Normalisation<br>MaxPool1D: 3*1</td><td>*2</td></tr>
<tr><td colspan="2">Flatten, Dimension_expand</td></tr>
<tr><td rowspan="6">Back End Network</td><td>Conv2D: 7*256*512<br>Batch Normalisation</td><td>L1;</td></tr>
<tr><td>Conv2D: 7*256*512<br>Batch Normalisation</td><td>L2;</td></tr>
<tr><td>Add (L1, L2)</td><td>L3;</td></tr>
<tr><td>Conv2D: 7*256*512<br>Batch Normalisation</td><td>L4;</td></tr>
<tr><td colspan="2">Add (L3, L4)</td></tr>
<tr><td colspan="2">Global Pooling;<br>Dense(feature embedding layer): 50<br>Dense: num_para</td></tr>
<tr><td colspan="3">Output: Parameters</td></tr>
</table>

Table 5.2: Model summary for the waveform structure, i.e. Model 2. It can be separated to front-end and back-end network, where the front-end is a combinations of sample level 1D Conv layers, and the back-end is two layers of residual layers.

illustrated in Figure 5.3. This model structure used Mel-spectrogram as input. It applies six different kernel shapes with 2D convolutional layers, four different kernel shapes with 1D convolutional layers, and concatenate the outputs of all ten layers together as input to the back-end network. The back-end network has two residual convolutional layers and two dense layers, which is the same as in Model 2.

### 5.2.4 Dataset description

In these experiments, the datasets are generated from Apple Loops[1], 64 guitar loops and drum loops respectively. They are compressed using different DRC parameter settings and the following datasets are generated:

For $DS_1$ to $DS_4$, the audio files are compressed with only one parameter changing while the others are kept the same. For $DM_1$ and $DM_2$, each audio file is compressed by two parameters changing at the same time. Take $DM_1$ as an example, each audio file

[1] `https://support.apple.com/kb/PH13426`

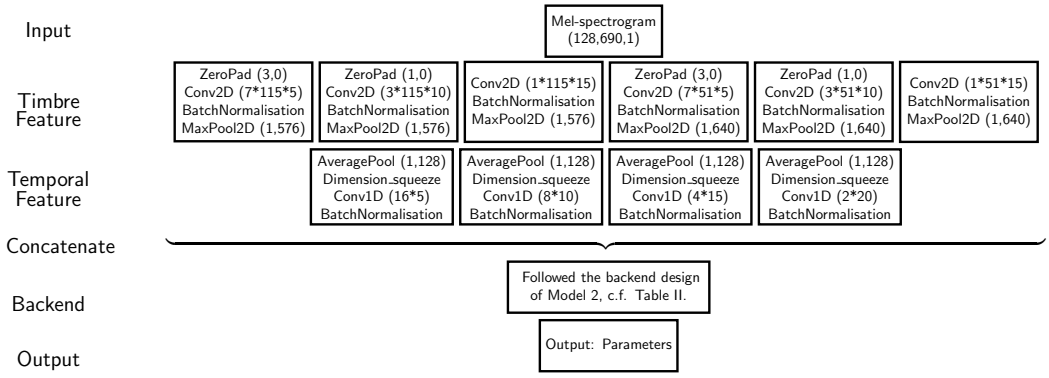



Figure 5.3: Model summary for the multi-kernel structure, i.e. Model 3. The front end network concatenates 11 Conv layers with different kernel shapes. The back end network is two layers of residual layers, which is the same as Table 5.2.

| | dataset generation | dataset size |
|---|---|---|
| $DS_1$ | $\theta$: 0 to 49dB with step of1dB | guitar: 65*50; drum: 64*50 |
| $DS_2$ | $\gamma$: 0 to 20 with step of 0.4 | guitar: 65*50; drum: 64*50 |
| $DS_3$ | $\tau_a$: 1 to 100ms with step of 2ms | guitar: 65*50; drum: 64*50 |
| $DS_4$ | $\tau_r$: 10 to 1000ms with step of 20ms | guitar: 65*50; drum: 64*50 |
| $DM_1$ | $\theta$: 10 to 47.8dB with step of 0.6dB<br>$\gamma$: 1 to 19.9 with step of 0.3 | guitar: 65*64; drum: 64*64 |
| $DM_2$ | $\tau_a$: 1 to 95.5ms with step of 1.5ms<br>$\tau_r$:10 to 955ms with step of 15ms | guitar: 65*64; drum: 64*64 |

Table 5.3: Dataset details for two instruments

is compressed by 8 threshold settings and 8 ratio settings. This process will produce 64 compressed audio for each raw audio loop. For example, $guitar_1$ is compressed using $\theta$ [10.0dB, 14.8dB, 19.6dB, 24.4dB, 29.2dB, 34.0dB, 38.8dB, 43.6dB], $guitar_2$ is compressed using $\theta$ [10.6dB, 15.4dB, 20.2dB, 25.0dB, 29.8dB, 34.6dB, 39.4dB, 44.0dB], and so on. For each audio file, the compression threshold grid is 4.8dB, but the combined set yields a finer grid, 0.6dB as noted in Table 5.3.

In the following subsections, models for one and two parameters are trained and tested using these datasets. A more complex dataset is generated and tested in Section 5.4.

### 5.2.5 Evaluation of different model designs

All models are trained using a small batch size of 8, and 15% of the data are used as validation set. The validation error is monitored to avoid overfitting to the training data

using an early stopping function. The training will stop when the validation error starts to increase. After the model is trained, the feature embedding is generated using this model. The feature embeddings are then used to train a random forest regression model, where the data is split into 80% training and 20% testing. The process is repeated 50 times randomly. The averages of the test prediction mean absolute errors (MAE) are reported in Table 5.4.

The DNN models are able to produce similar results for $\theta$ and $\gamma$. However, the prediction error of $\tau_a$ and $\tau_r$ are relatively higher compared to handcrafted features. Because the audio materials are not preprocessed to emphasise any DRC's effect, the model would react better when a specific DRC parameter has a more significant influence on the raw audio, i.e. $\theta$ and $\gamma$. Moreover, the handcrafted features are tuned specifically to extract information from a temporal region of the audio where a certain parameter is the most effective. Comparing the two types of models, using raw audio as input provides better performance in 6 out of 8 cases for single parameter prediction. Many reasons can lead to this result. First of all, the frame size this research used for the Mel-spectrogram is 512, which is considerably large given that this Thesis' problem focuses on small transient times. Secondly, Mel-spectrogram works well when researchers are aiming at retrieving high level music information, but it may smear useful spectral information that is important in this problem. The improvement of the model structure and hyperparameters are given Section 5.3.1-5.3.3.

| | | $\theta$ | $\gamma$ | $\tau_a$ | $\tau_r$ |
|---|---|---|---|---|---|
| Guitar | Model 1 | 1.781dB | 0.657 | 4.338ms | 35.589ms |
| | Model 2 | 1.206dB | 0.751 | 3.192ms | 32.893ms |
| | Model 3 | 1.034dB | 1.009 | 3.273ms | 71.288ms |
| | Handcrafted | **0.903dB** | **0.623** | **0.845ms** | **10.442ms** |
| Drum | Model 1 | 2.994dB | 0.961 | 3.829ms | 58.394ms |
| | Model 2 | 2.627dB | 0.932 | 3.480ms | 43.668ms |
| | Model 3 | 2.953dB | 1.218 | 7.694ms | 93.064ms |
| | Handcrafted | **0.915dB** | **0.655** | **1.194ms** | **12.714ms** |

Table 5.4: Prediction MAE for the regression model using feature embeddings learnt from each DNN as well as handcrafted features, when predicting individual parameters of DRC.

The results are not encouraging when using the models trained to predict individual DRC parameters. This might due to the scenario is simple and using a DNN is over complicated and causing overfitting. The size of the models are relatively big. Model 1 has 60k parameters, and both Model 2 and 3 has over 5 millions parameters, which may be a too complex model to the problem. In contrast, improvements are shown when

| | | $\theta$, $\gamma$ | $\tau_a$, $\tau_r$ |
|---|---|---|---|
| Guitar | Model 1 | 1.854dB, 0.529 | 2.081ms, 20.184ms |
| | Model 2 | 1.666dB, **0.460** | 1.725ms, 18.357ms |
| | Model 3 | 1.567dB, 0.618 | **1.565ms**, **15.588ms** |
| | Handcrafted | **0.912dB**, 0.883 | 2.100ms, 25.079ms |
| Drum | Model 1 | 1.810dB, 0.800 | 5.061ms, 27.005ms |
| | Model 2 | **1.112dB, 0.391** | 4.506ms, **13.609ms** |
| | Model 3 | 2.170dB, 0.782 | 6.463ms, 21.427ms |
| | Handcrafted | 3.233dB, 0.684 | **2.354ms**, 14.980ms |

Table 5.5: Prediction MAE for the regression model using feature embeddings learnt from each DNN as well as handcrafted features, when the model predicts two DRC parameters jointly.

training the model to learn two parameters simultaneously. The training data for the two parameters model are $DM_1$ and $DM_2$, which has a larger size than the single parameter model training dataset. The results are shown in Table 5.5. Comparing with Table 5.4, the model produces better results, especially for release time. For $\theta$, $\gamma$, and $\tau_r$, the DNN model is able to yield a better performance than handcrafted features. Model 2 is the best performing model in this experiment. This model provides the best performance for 4 out of 8 cases. For reference, the range of each parameters are 49dB for $\theta$, 19 for $\gamma$, 99ms for $\tau_a$ and 999ms for $\tau_r$. The larger range of release time resulted in higher prediction error compared to the other parameters.

## 5.3 Model Tuning

In the previous section, several model designs from the literature has been explored. In this section, the author will tune the structures to fit the Thesis' problem better, along with the analysis of the model components' impact on the result. The following three subsections discuss the improvement of each model.

### 5.3.1 Improvement on Model 1

This section is aiming at improving the performance of Model 1. There are several reasons to use time-frequency representation as input, including that the data size is reduced, the ability to control the frequency and time resolution is increased, and using 2D convolution will give users more choices of kernel size.

The previous settings for the hyperparameters are directly taken from the literature [Pons et al., 2017b, Lee et al., 2017]. Here the network structures and hyperparameters

will be optimised to fit this particular problem better. As it is explained in the Chapter 2, many factors will influence the model's performance. In this section there are three factors considered: whether using Mel-spectrogram or spectrogram is benificial, the kernel shape of the model, and the time frame length for STFT. One assumption in this section is that the Mel-spectrogram smeared useful information and led to poor performance, therefore, using spectrogram alone can be assumed to improve the performance. Due to the fact that this Thesis' problem requires focussing on a short transient time at some point, as well as the sample-level Model 2 shows a better performance in the previous experiment, it can be assumed that a short time frame, i.e. a better time resolution, and a smaller kernel size will improve the performance. Based on these two assumption, the following experiments are conducted.

Since changed experiments are designed to improve the model, there is no need to run a thorough experiments for all the datasets. This section uses the drum dataset: $DS_3$, $DS_4$ and $DM_2$, and the predicted attack/release time errors are reported. The first experiment aims to select the most suitable input signal format. The same frequency resolution is applied for both representations. The prediction error shows a large improvement in Table 5.6. Especially the prediction error of release time reaches a similar level performance with Model 2. All cases exceed the performance of the original setting of Model 1 as well. It can be concluded that spectrogram will be a more suitable representation for this model and this problem.

| Para(ms) \ Input signal | Melgram | Spectrogram |
|---|---|---|
| $\tau_a$ | 3.829 | **2.415** |
| $\tau_r$ | 58.394 | **33.085** |
| Joint prediction | 5.061<br>27.005 | **4.656**<br>**17.781** |

Table 5.6: Melgram vs Spectrogram, prediction performance when changing input representations.

In the second experiment, the time frame length for spectrogram is investigated. The results in Table 5.7 provides a clear trend showing that with the decrease of the time frame length, the prediction error drops as well. This experiment did not progress the experiment with frame sizes smaller than 128 samples because this scenario may be similar to using raw audio input.

The third experiment is conducted while altering the kernel size of the model. The

| Para (ms) \ Time frame length | 512 | 256 | 128 |
|---|---|---|---|
| $\tau_a$ | 2.415 | 2.197 | **2.042** |
| $\tau_r$ | 33.085 | 30.546 | **26.698** |
| Joint prediction | 4.656<br>17.781 | 4.271<br>16.846 | **3.141**<br>**15.752** |

Table 5.7: Prediction performance when changing frame size of the input spectrogram

original design uses five 2D convolutional layers with 3 by 3 kernel. To capture the audio features in multiple feature dimensions, the kernel sizes and combinations are altered. The experiment reduces the depth of the 2D layers and increases the depth of 1D convolutional layers at the same time. Except for the release time results, the rest of the performances did not show significant improvement, c.f. Table 5.8. For simplicity, the five convolutional layers with 3*3 kernels will be kept in further experiments.

| Para (ms) \ Kernel size | 5(3*3) | 4(3*3)+1(1*3) | 3(3*3)+2(1*3) |
|---|---|---|---|
| $\tau_a$ | 2.042 | 1.966 | **1.962** |
| $\tau_r$ | 26.698 | 21.173 | **18.491** |
| Joint prediction | **3.141**<br>15.752 | 4.232<br>**14.436** | 3.941<br>16.791 |

Table 5.8: Kernel shape changes for Model 1, with different combinations of 2D and 1D Convolutional layers

### 5.3.2 Improvement on Model 2

Followed by the previous section, tuning the structure and hyperparameters of Model 1 leads to a performance improvement. In many cases, it exceeds the performance of using raw audio signal as input. In this section, the author explores the improvement for Model 2. One conclusion can be drawn from the previous experiment is that having large filter size in the time axis will result in poor performance in predicting $\tau_a, \tau_r$. Smaller filter size have also been tested on prediction $\gamma, \theta$, but the improvement is not as significant as the temporal parameters, therefore this section did not present this result. Apart from the size of the filters, the author also alters the number of filters and layers of the network. The prediction errors are provided in Table 5.9. However, the results have not improved as significantly as they do for Model 1. The advantage of the sample level CNN is having a fine resolution in time domain, tuning the CNN model and its input time-frequency

representation can counterbalance the differences and provide a decent prediction result.

| | Initial | ↑ Filter size | ↓ Filter number | ↓ Layers |
|---|---|---|---|---|
| $\tau_a$ | 3.480ms | 7.772ms | **3.227ms** | 3.784ms |
| $\tau_r$ | 43.668ms | 48.834ms | **39.740ms** | 45.030ms |
| Joint prediction | 4.506ms<br>**13.609ms** | 6.683ms<br>20.486 | 5.847ms<br>17.622ms | **3.864ms**<br>16.619ms |

Table 5.9: Tuning results for Model 2, when the author increases filter size, reduce filter numbers, and reduce layers respectively. The final hyperparameters used are: filter size: 5; filter number: 32; and 4 layer.

### 5.3.3 Improvement on Model 3

As it is mentioned in Section 5.2.3, the multi-kernel structure that has proved to be efficient for music tagging problem [Pons et al., 2017b] is also considered. Since the DRC may impact audio events in the short and long-term and impact different frequencies in a non linear fashion, it makes sense to use multiple kernels at the same time. The prediction results for single model and joint model are given in Table 5.10.

| | Model 1 tuned | Model 3 | Model 3 tuned |
|---|---|---|---|
| $\theta$ | 1.543dB | 2.953dB | 2.602dB |
| $\gamma$ | 0.746 | 1.218 | 1.184 |
| $\tau_a$ | 2.024ms | 7.694ms | 7.691ms |
| $\tau_r$ | 26.698ms | 93.064ms | 41.678ms |
| Joint model | 1.019dB<br>0.417 | 2.170dB<br>0.782 | 1.351dB<br>0.394 |
| | 3.141ms<br>15.752ms | 6.463ms<br>21.427ms | 3.644ms<br>17.688ms |

Table 5.10: Improvement for Model 3, with spectrogram and a reduction of window size

The results for Model 3 are not as good as the best performance in Section 5.3.1. Based on the conclusion from the previous sections, the performance of Model 3 is improved by using spectrogram with a small time frame size as input, as well as decreasing the kernel size for the temporal features. These changes do not improve the prediction error rate significantly however. Model 3 combines multiple feature representation layers, therefore, the trainable parameters are much more comparable to Model 1 and Model 2. The small performance improvement may due to the complexity of this model. It might also be because of the shallowness of this model. The model concatenated multiple layers together,

but they are all single layers, therefore the depth of the model is 3, which compared to the other model design is very shallow. A deeper model can be considered in future work.

## 5.4 Evaluation on simultaneous parameter estimation and polyphonic music data

In this section, the dataset is extended from only two parameters changing at a time, to all four parameters changing simultaneously. The process is done in the same way as described by Table 5.3. Changing four parameters together means a substantial growth of the data, therefore, instead of 8 settings for each parameters, the amount of settings is reduced to 5 in this case, i.e. $drum_1$ is compressed using $\theta$: [10.0dB, 18dB, 26dB, 34dB, 42dB], $\gamma$: [1.28:1, 5.12:1, 8.96:1, 12.80:1, 16.64:1], $\tau_a$: [1ms, 21ms, 41ms, 61ms, 81ms], and $\tau_r$: [10ms, 210ms, 410ms, 610ms, 810ms]; $drum_2$ is compressed using $\theta$ [11.0dB, 19dB, 27dB, 35dB, 43dB], $\gamma$: [1.76:1, 5.60:1, 9.44:1, 13.28:1, 17.12:1], $\tau_a$: [3.5ms, 23.5ms, 43.5ms, 63.5ms, 83.5ms], and $\tau_r$: [35ms, 235ms, 435ms, 635ms, 835ms], and so on. The dataset details are outlined in Table 5.11.

| | dataset generation | dataset size |
|---|---|---|
| D4P | $\theta$: 10 to 49dB with step of 1dB<br>$\gamma$: 1.28 to 20 with step of 0.48<br>$\tau_a$: 1 to 98.5ms with step of 2.5ms<br>$\tau_r$:10 to 985ms with step of 25ms | drum: 64*625 |

Table 5.11: Dataset details for data generated by changing four parameters together

The results are compared between the prediction of the regressor trained on handcrafted features and the feature embedding learnt by Model 1, c.f. Table 5.12. The predictions of the four parameter model are not as good as the two parameter ones. However, the model trained on four parameters shows its advantage when compared to handcrafted features. When several attributes of the audio are changing at a time, handcrafted features, that are designed to measure specific attributes like attack time differences, tend to lose their benefit compared to a neural network. The MAEs for attack and release time have grown, but as a reference, the range of the two parameters are 99ms vs. 999ms. The percentage of the prediction error over parameter range is also outlined in Table 5.12.

The author is also interested in testing the model using more complex audio materials, that is, polyphonic music. Fifty audio segments are randomly selected from the mixed

| | Handcrafted features | Feature embeddings |
|---|---|---|
| $\theta$ | 2.937dB / 7.34% | 2.369dB / 5.92% |
| $\gamma$ | 3.447 / 17.24% | 3.265 / 16.33% |
| $\tau_a$ | 13.926ms / 14.07% | 10.868ms / 10.98% |
| $\tau_r$ | 120.577ms / 12.07% | 79.045ms / 7.91% |

Table 5.12: Prediction MAE when using handcrafted features and feature embeddings on large scale dataset, when predicting four DRC parameters. The percentage of the predicted error over parameter range is also outlined

music of the MedleyDB dataset [Bittner et al., 2014]. The amount of 50*625 compressed audio signals are generated using the same method described in as Table 5.11. In this experiment, the random forest regression model trained by drum loops' features is used to predict the compression parameters of the polyphonic audio. The two types of features are also handcrafted features and the feature embeddings from the best performing DNN model. The predicted MAEs are reported in Table 5.13. The results from the model trained by feature embeddings are obviously better. This result is reasonable because when researchers train the DNN model, the twin model is designed in the way that the feature embedding would focus on the difference between the two input audio signals. Meanwhile the handcrafted features are highly depended on audio content. It is surprising that the prediction MAE for mix audio is better than drum loops, when comparing the last column of Table 5.12 and Table 5.13. This result might be explained by an assumption that having a richer audio content provides a benefit for the DNN model. The results also show that the model is still able to provide a decent prediction when using data from another dataset that was not considered during the model design and was not used for training at all. This indicates improved robustness and generalisation of the assessed method.

| | Handcrafted features | Feature embeddings |
|---|---|---|
| $\theta$ | 11.585dB | 1.697dB |
| $\gamma$ | 5.104 | 2.194 |
| $\tau_a$ | 26.628ms | 9.873ms |
| $\tau_r$ | 268.420ms | 160.629ms |

Table 5.13: Prediction MAE for mixed audio whose feature embeddings are generated using DNN model trained by drum loops

## 5.5 Conclusion

In this chapter, the author proposed several DNN model designs to extend the feature extractor in the proposed intelligent DRC control system. Handcrafted features failed to provide good prediction when trying to predict several parameters together. DNN models start to show their advantages compared to handcraft features when predicting more than one DRC parameter. The improvement becomes substantial when predicting all four parameters together, which is encouraging as it implies the DNN model would fit the real world scenario better. Across the model designs, the CNN model using a high temporal resolution spectrogram as input provides the best performance.

In this chapter, it was discovered that the performance would improve significantly when an appropriate time-frequency representation is applied as input, for example to use wavelet method. The multi-kernel model does not provide the best performance in the experiments. This might be because of the complexity of the model, i.e., a large network with more parameters to train, possibly requires more training data. However, it is still worth considering to use the multi-resolution time-frequency representations as input. When increasing the complexity of the model, i.e. predicting several parameters together, the performance to predict two parameters are much better than only train it to predict one. However, the performance for four parameters drop substantially compared with predicting only two parameters. One explanation could be that the grid size are larger when training data for the four parameters are generated. The results may improve using a finer grid and consequently the increased size of the training dataset.

In conclusion, the DNN model provides the ability to train one generic model for all parameters of an audio effect. It helps to reduce the limitations of the handcrafted features approach. Further research includes optimisation on the designed intelligent control model, which is presented in the following chapters.

# Chapter 6

# Audio Similarity Model for the Perceptual Aspects of Sound Modified by Audio Effects

An innovative system design for the intelligent control system for DRC had been proposed and developed from Chapter 3 to Chapter 5. From this chapter onwards, the focus is moving to the optimisation and evaluation of the proposed system. Chapter 6 proposes an audio similarity model that is targeting the audio perceptual aspects of sound affected by DRC. An optimisation algorithm is proposed in Chapter 7. A subjective evaluation that can serve as a perceptual support of the intelligent control model is presented in Chapter 7.

This chapter starts with an overview of music similarity as well as the speciality of the similarity this Thesis requires. Section 6.1 also includes the motivation of the similarity model designed in this chapter. This model is derived from the method proposed in Section 3.2.2. The model has been decomposed into several components and the design of each component is provided in Section 6.2 and developed and evaluated in Section 6.3. The overall evaluation and conclusion are followed at the end of this chapter in Section 6.4 and 6.5.

## 6.1 Motivation

As discussed in Section 2.4.2, similarity assessment in music ranges from relatively simple comparison between the sounds of instruments, through the assessment of complex music theoretical concepts like melody, rhythm and structure, to matching high-level semantic labels like genre and moods. Perceived similarity between two pieces of music results from a combination of these aspects, including factors related to music performance and audio production.

With the exception of work by Tardieu et al. [2011], relatively little attention has been paid to the effects of production style and audio processing within the body of works on music similarity and classification. The research in this chapter is attempting to fill this gap by examining how audio effects transform sound and how their effect on different perceptual attributes may be modelled.

Understanding and modelling how signal transformations introduced by audio effects are perceived may help creating intelligent audio production tools, including audio effects configured using descriptive terms or by the use of sound examples. Estimating similarity between sounds or music pieces plays a crucial role in these applications, as well as in broader areas of music informatics such as audio retrieval and recommendation systems.

Most audio effects can be modelled mathematically, however the perceptual impact of them is beyond the scope of their mathematical representations. This chapter focuses on the perceptual aspects of the dynamic range compressor. The mathematical approach to analyse the compressor parameters is not within the scope of this research. Most audio effects modify several perceptual attributes of sound, especially in the case of non linear effects [Wilmering et al., 2013], it is therefore sensible to define success criteria using a measure of audio similarity.

Defining a suitable similarity metric between sounds is a complex problem however, since similarity depends on a number of perceptual qualities, a common approach in music information retrieval is to use perceptually motivated audio features such as Mel-Frequency Cepstral Coefficients (MFCC) [Jensen et al., 2009], for instance, to quantify timbre similarity. This approach does not take auditory perception fully into account.

Auditory models incorporate the response of the outer, middle and inner ear and possibly more complex behaviour such as level dependence and non linearity in frequency sensitivity and intensity. Therefore they can be useful in quantifying audio similarity. Nu-

merous components have been proposed in auditory models, including different types of auditory filters [Lyon et al., 2010]. Their ability to characterise changes introduced by audio effects has not yet been assessed however.

This chapter focuses on the design and evaluation of a similarity model for DRC taking relevant perceptual factors into account. The model design process assesses several auditory model components and statistical modelling techniques commonly proposed in the literature [Moore, 2014, Allamanche et al., 2001, Shao et al., 2009, Qi et al., 2013, Hershey and Olsen, 2007], etc.

To assess individual model components, such as different types of auditory filters, a series of experiments are performed with controlled audio transformations. By controlled transformation, the author means a series of DRC parameter settings. The correlations between the output of the designed model and the parameter settings are examined. There are also experiments designed to cluster audio materials using the output of the model as a feature. The quality of the clusters can be used to inform the design of our proposed similarity model. We can assume audio examples of the same category or examples subject to similar processing will be an easier target for clustering algorithms using the proposed similarity measurement method, hence cluster quality can become a proxy for the quality of similarity assessment. A commonly used audio similarity measure [Jensen et al., 2009] is implemented as baseline to be compared with, albeit in different contexts than the original algorithm. The perceptual validity of the models is not tested in this chapter. The next chapter of the Thesis presents a listening test that looks instead at baseline human performance in identifying DRC related perceptual effects to compare with.

The proposed model can be used in training, evaluation or optimisation in intelligent DRC control methods. It also has applications in audio engineering and other music informatics tasks. For instance, a music producer may search for heavily compressed drum samples from a catalogue, a musician may want to find similarly sounding instrument samples, or a listener may want to find similarly mastered recordings or separate remastered tracks from earlier releases. The model design, development, and evaluation will be introduced in the next section.

## 6.2 Method

As outlined in Section 6.1, audio effects influence several perceptual aspects of sound including timbre, loudness and pitch. It is challenging to unify all factors in a single model. Therefore, the proposed model only considers the attributes modified by the DRC, i.e. loudness and timbre [Wilmering et al., 2013].

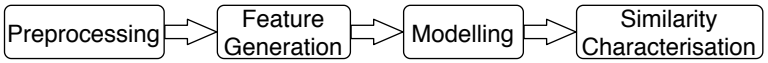



Figure 6.1: Components of the proposed similarity modelling method

The proposed method uses the computation steps illustrated in Figure 6.1. In order to derive perceptually relevant features, auditory models are considered during the process. Some parts of the auditory system can be described as a single filter, others are level dependent filter banks. The audio signals can be processed by the simulation filters resembling to the process within the auditory system. This is considered preprocessing in Figure 6.1. To model the non linear frequency resolution of the cochlea (see Section 2.4.1), researchers proposed several types of filter banks, including Mel-scaled banks, Bark-scaled banks, Gammatone family filter banks, etc. These differ in the distribution of center frequencies as well as the shape and other properties, such as symmetry, of the individual auditory filters [Lyon et al., 2010]. Therefore, different filter banks are used to compute representative features (see Section 6.2.2 for details) and these are compared afterwards. Subsequently, parametric statistical models are fitted on the features, and the distance or divergence between the distributions can be used to estimate similarity between sounds. The choice of fitting statistical models is due to the fact that timbre is a complex notion and need to be modelled along multiple dimensions. A simple method such as Euclidean distance between means or simple low-order statistics may not be able to capture the difference between complex multivariate and possibly multimodal distributions. A Gaussian Mixture Model is therefore more suitable in this scenario. The following sections illustrate the design details of our similarity model targeted at isolated instruments sounds.

### 6.2.1 Preprocessing - middle ear and loudness model

In general, auditory models contain the filter through the outer ear, filter through the middle ear, and model the excitation pattern which leads to the loudness pattern [Moore,

2014]. The first filter, from free field to out ear is related to the environment and the position of the sound source with respect to the ear, therefore, this filter is not considered in the proposed model. In this research, the author considers the filter from the outer to middle ear as well as loudness equalisation.

The middle ear response is evaluated as suggested in [Allamanche et al., 2001], which is also used in the similarity algorithm proposed in [Pampalk et al., 2008]. The filter response is defined in Equation 6.1. The response curve is given in Figure 6.2(a). This shows that the filter has a boost in the frequency range between 2-4kHz and reduces the signal amplitude significantly at low and high frequencies.

$$H_{dB}(f_{kHz}) = -3.64 \times f^{-0.8} + 6.5 \times exp[-0.6 \times (f - 3.3)^2] - 10^{-3} \times f^4 \tag{6.1}$$

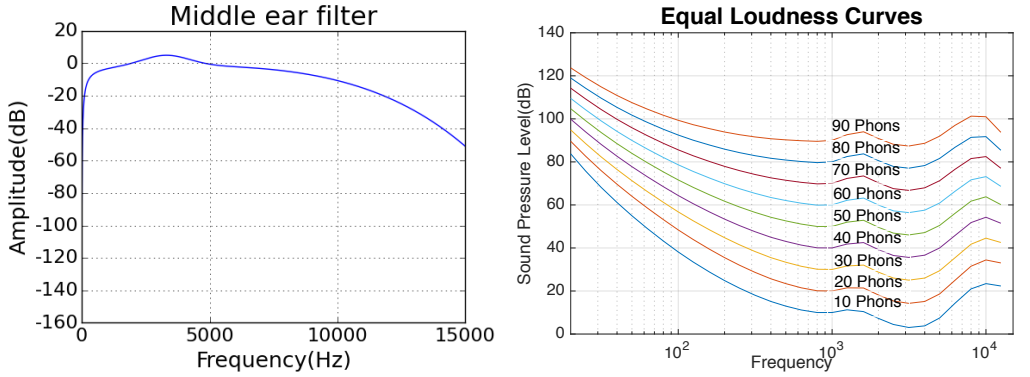
Loudness, as a perceptual correlate of sound intensity, is also in the focus of this work [Moore and Glasberg, 1997]. A loudness model is useful in designing a feature representation that is relevant to perception. There are multiple components involved in a loudness model too. In the preprocessing stage, we consider using equal loudness curves, c.f. Figure 6.2(b). There is little consensus about the transformation of neural patterns into perceptual measures [Röhl and Uppenkamp, 2012], therefore neural pattern encoding of the audio will not be considered.

Robinson and Dadson [1956] introduces the equal-loudness relations for pure tones in free-field conditions. This curve and its improvement are also included in the loudness model illustrated in [Glasberg and Moore, 2006]. Equal-loudness contours are also published in the ISO226 standard. These contours are plotted in Figure 6.2(b).

### 6.2.2 Feature extraction and timbre modelling

A primary concern in this part of the model is to represent spectral and temporal characteristics of sounds closely corresponding to *timbre changes* induced by audio effects. This is one of the perceptual aspects of sound modified by the DRC [Wilmering et al., 2013]. From an auditory perspective, this subsection considers the inner ear response and various auditory filter banks.

Timbre is a multidimensional, subjective and context dependent phenomenon, therefore it is not easy to provide a clear definition. A relatively widely accepted description is




(a) Middle ear filter impulse response

(b) Equal loudness curve, generated from stander ISO 226

Figure 6.2: Two aspects of the auditory model that are considered in the preprocessing stage

given in [American_Standards_Association and Acoustical_Society_of_America, 1960]: *Timbre is the quality that distinguishes sounds with the same pitch, loudness, and duration.* Related features can be divided into spectral/temporal features or static/transient features. In the frequency domain, MFCCs can be used as non linear spectral scaled features to represent timbre information. Spectral centroid and higher order statistical features are often used as well [Peeters et al., 2011], while more recently Neural Networks have also been proposed to model timbre [Pons et al., 2017a]. The focus at this stage of this work is on conventional features for better understanding of the impact of signal components and perceptual models.

Changes in the energy envelope can provide timbre information in the time domain. To incorporate temporal information into the model, the use of delta features that correspond to the changes in the energy curve are considered. These features can then be expected to represent timbre changes influenced by attack and release times of audio effects. These are important parameters in DRC designs [Giannoulis et al., 2012] and are expected to impact transient sounds the most.

After applying the middle ear filter and equal loudness patterns to preprocess the audio signal, it is reasonable to consider the model of the cochlea, c.f. Section 2.4.1. In related similarity model designs, a popular feature of choice is MFCC features (cf. Aucouturier and Pachet [2002], Heittola et al. [2009], Logan and Salomon [2001]). However, this work aims to incorporate and characterise the response of the human auditory system more faithfully. It is reasonable to consider more accurate biologically inspired alternatives to MFCCs, for example, Gammatone cepstral coefficients [Shao et al., 2009].

The Gammatone family of filters [Shao et al., 2009, Qi et al., 2013] are increasingly used for this purpose. To improve simulation and provide a better fit to psychophysical data compared to previous approaches [Unoki et al., 2006], the Gammachirp filter was introduced by Irino and Patterson [1997] as a theoretically optimal auditory filter derived from the Gabor function, which is know to be able to achieve minimum uncertainty in a joint time-scale representation [Gabor, 1946]. The Gammachirp filters also have a sharp drop in the high frequency part compared with Gammatone filters, which are able to reflect the temporal masking effects of the auditory system better. Equation 6.2 defines the impulse response of Gammatone filters, where $b = 1.019 * ERB(f) = 1.019 * (24.7 + 0.108f)$ [Slaney, 1993]. The implementation of this filter bank consists of 8th order linear (IIR) filters corresponding to a bank of 4th order Gammatone filters centred around ERB critical band frequencies. The Gammachirp filter is described in Equation 6.3, where $\sigma$ is a time constant and $\phi$ is the phase. In both equations, $f$ represents the center frequencies placed along the ERB scale. In this research, the experiments apply and compare the following three types of auditory features: MFCC, Gammatone and Gammachirp derived features.

$$g(f,t) = t^3 exp(-2\pi bt) cos(2\pi ft) \tag{6.2}$$

$$g(f,t) = exp(-t/2\sigma) cos(2\pi(ft + c/2t^2) + \phi) \tag{6.3}$$

All time domain signals (e.g. filter bank outputs) are processed using short-time windows, with a frame size of 1024 samples (23.22ms) and a hop size of 64 samples (1.45ms). The time resolution in this research is at the millisecond level for both attack time and release time. The window and hop sizes are chosen correspondingly. The time domain input audio signals are processed by Gammatone or Gammachirp filter banks, and the log energy within each time frame and for each frequency band is calculated. For energy compacting and decorrelation, the Discrete Cosine Transform (DCT) is applied to the sub-band energy signals of the Gammatone and Gammachirp filters. The type II DCT is used for its favourable computational and energy compaction properties [Rao and Yip, 2014]. There are 40 frequency bands ranging from 0-22050Hz in all three types of filter banks. This also helps to keep consistency across three types of auditory filter models we compare and follows the procedure of the typical MFCC calculation (e.g. [Jensen et al., 2009]), once band energies are computed. We take the same amount of coefficients as the output of typical MFCC features, 13, due to the fact that the coefficients larger than 13 are relatively

small in value while it is also a commonly accepted limit for such processing.

Another design consideration is the application of delta features, i.e., first and second order differences between samples. These features are able to characterise dynamic information in the similarity model [Qi et al., 2013]. In this experiment, the features using the first and second order derivatives are calculated as shown in Equation 6.4 and 6.5,

$$\Delta F(n,u) = \frac{\sum_{k=1}^{K} k(F(n+k,u) - F(n-k,u))}{2\sum_{k=1}^{K} k^2}, \tag{6.4}$$

$$\Delta\Delta F(n,u) = \frac{\sum_{k=1}^{K} k(\Delta F(n+k,u) - \Delta F(n-k,u))}{2\sum_{k=1}^{K} k^2}, \tag{6.5}$$

where $n$ is the time domain sample, $u$ is the frequency band, and $K$ is the offset of the time sample. More implementation details will be provided in Section 6.3.2. In [Krishnan et al., 2013] authors proposed a dynamic feature computation method based on Savitzky-Golay (S-G) filter. This filter is normally used for data smoothing while it preserves the peak shape as well as high frequency components. The implementation of the S-G filter can be as simple as using polynomial approximation of an impulse sequence. The impulse sequence is independent from the signal, therefore, it can be calculated beforehand. This smoothing filter is tested along with auditory features and dynamic features in Section 6.3.2.

### 6.2.3 Statistical Modelling

At this stage, after the computation of features, a parametric statistical model is used to characterise a sequence of feature vectors. All representations, regardless of the filter bank considered, produce a matrix $\mathbb{R}^{N\times T}$, where $N = 13$ is the number of coefficients and $T$ is the number of time frames. To facilitate the estimation of similarity between these representations, we use parametric statistical models. However, typical audio signals do not necessarily have a known parametric distribution.

To address this issue, many previous works suggest using Gaussian Mixture Models (GMM), see e.g. [Jensen et al., 2009]. Most music signals do not necessarily follow a normal distribution [Arora and Kumar, 2014] but a number of normal mixture components can capture complex distributions, with well understood and readily available methods to estimate their parameters, such as Expectation Maximization [Bilmes, 1998]. Therefore a GMM is used with several components, which will be able to provide a more precise approximation of audio signals. An example of histogrammed features is provided in Figure

6.3. For the purpose of visual inspection, a violin loop from the RWS instrument database [Goto et al., 2003] is used. This is processed using Gammachirp filter banks as described in Section 6.2.2. Histograms of 6 of 13 coefficients are illustrated in Figure 6.3. Empirical observations suggest that using 2-3 Gaussian components in this modelling stage is appropriate.

To be rigorous however, the performance of different hyper-parameters is tested experimentally as well. For instance, the exact number of Gaussian components and the type of the covariance matrix are tested, analysed, and selected in Section 6.3.3.

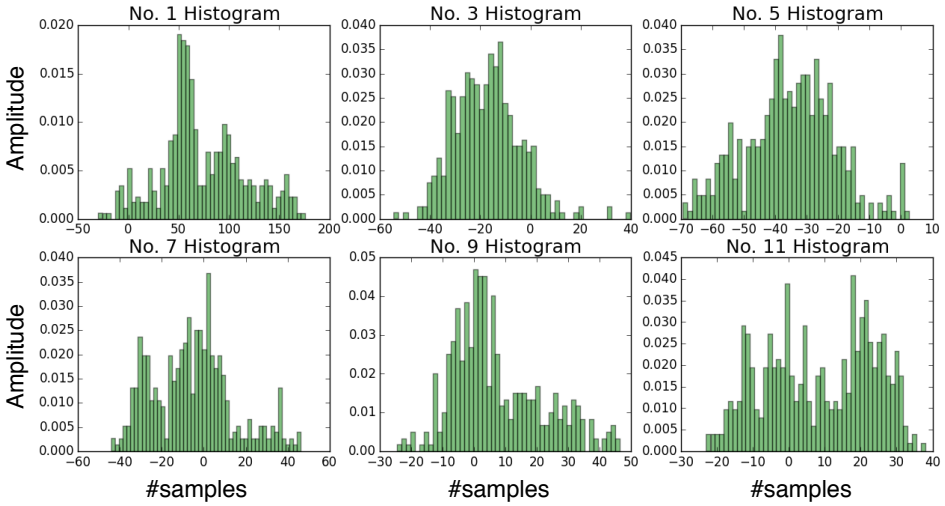



Figure 6.3: Histograms for DCT coefficients of Gammachirp subband signals for a short violin sound example.

### 6.2.4 Similarity Estimation Between Statistical Models

A commonly used method to estimate the similarity between different distributions is calculating the divergence (see Section 2.4.2). The Kullback-Leibler (KL) divergence is often used to measure the information gain between a probability model when compared to a reference. This divergence is chosen over other methods due to the observation that the Kullback-Leibler divergence performs well in similar music information retrieval tasks [Aucouturier and Pachet, 2002]. Its good performance is crucially linked to its value asymptotically going towards infinity when one of the distribution goes towards zero density [Jensen et al., 2009]. It is often applied to compute the divergence between Gaussian distributions for the measurement of the information loss when using one distribution to approximate

the other. Alternative methods include the Jensen-Shannon divergence [Fuglede and Topsoe, 2004], which can be considered for evaluation in future work. The model introduced in the previous section involves a GMM with several Gaussian components. The divergence between mixtures of Gaussian models is not analytically tractable, therefore the approach proposed in [Hershey and Olsen, 2007] is applied which is based on variational Bayes approximation.

In the next section, each processing component of the proposed similarity model is assessed in a series of tasks relevant to the similarity estimation of processed sounds.

## 6.3 Model Development

This section examines the design considerations discussed in Section 6.2. As baseline, the first experiment considers a model similar to [Jensen et al., 2009], where the auditory filters are omitted, 40-frequency bands, 13 coefficient MFCCs are used as features, with two-components GMM and KL divergence as parametrisation and similarity measure respectively. This is depicted in Figure 6.4.

In the following experiments, the model components will be replaced individually and the performance will be compared with the baseline system. The final model design will be decided based on the results. In the implementation of both Mel scale filter banks and Gammatone filter banks, 40 frequency bands are used as mentioned in Section 6.2.2. The features are calculated in a frame-wise manner, with the frame size of 1024 (23.22ms) samples, 64 (1.45ms) hop size and smoothed with a Hanning window.

The overall performance of the final model will be examined in Section 6.4.

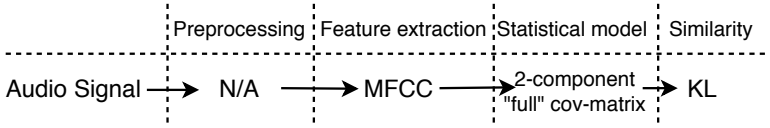



Figure 6.4: Baseline system components

### 6.3.1 Preprocessing

To assess if the auditory filters discussed in Section 6.2.2 can improve similarity estimation, there will be two experiments performed. Different preprocessing filters are applied to the audio signals in the similarity model. Audio signals can be classified by either instrument

types or audio effects applied on them. The experiment in this section uses the output of the similarity model as feature, and uses it to classify audio materials. The inter and intra class similarity algorithms [Manning et al., 2010] can be performed to evaluate the quality of the classifier and therefore the efficiency of the similarity model. Two sets of audio signals are selected and generated. The first one includes different instrument loops, and second one contains the loops from the same instrument with different audio effects applied to them. The timbre difference between instruments are relatively significant, therefore, the second set of audio signals generated focuses on minor timbre changes. It is assumed that applying auditory filters will improve the results for both tasks.

The proposed model settings are provided in Figure 6.5 where the model uses both auditory filters individually and jointly. All the results are also compared with the baseline system in Figure 6.4. Additional components i.e. using the distribution of Gammatone features modelled with 2-component GMMs are also included. The reason of choosing Gammatone filters and 2-component GMMs is justified using empirical observations discussed through experiments detailed in Section 6.2.3. The experiments described in this section aim for a more precise choice of the preprocessing filters.

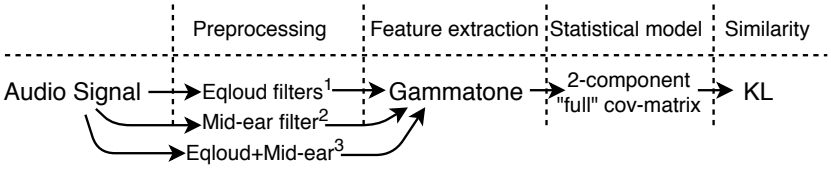



Figure 6.5: Assessment of middle and inner ear responses. $1-3$ represents three types of combinations of preprocessing filters.

Inter and intra class similarity is a notion that is widely used to verify the quality of classification or clustering tasks [Bezdek and Pal, 1998]. Since the goal of this chapter is to design a similarity model, it might not be a very strong argument to run classification tasks to illustrate the performance of a similarity model. Instead, the author chooses to evaluate the cluster quality based on the similarity results. In this experiment, the similarity across guitar loops as well as between guitar and another 7 instruments loops are calculated. For each type of instruments the author selects 40 loops from AppleLoops[1] where it can be assumed the recording condition is good and consistent. The 8 instruments are guitar, piano, bass, violin, accordion, cello, dizi, and drum. All loops are loudness normalised to $-23dB$ to exclude the effect of loudness in this case. The similarity model outputs a 1D

[1] `https://support.apple.com/kb/PH13426?locale=en_US&viewlocale=en_US`

signal for each audio pair. One of each pair is always a guitar loop. Since ground truth of the other instrument is known, a pseudo_F index can be performed to evaluate if the similarity can be a good feature to classify the instruments. The pseudo_F index describes the ratio of between class variance to within-class variance. It can be used as a quality measure for clustering or classification [Caliński and Harabasz, 1974]. It can be calculated as shown in Equation 6.6, where $GSS$ is the inter class sum of squares, and $WSS$ is the intra class sum of squares. $N$ is the number of data points, and $K$ is the number of classes. In this case, $N = 320$ and $K = 8$. The same procedure will be applied to drum loops.

$$Psuedo\_F = \frac{GSS/(K-1)}{WSS/(N-K)} \tag{6.6}$$

| Model Type | | Performance (Pseudo_F) | |
|---|---|---|---|
| | | Experience I | Experience II |
| Baseline System | Guitar | 19.462 | 3.670 |
| | Drum | 143.913 | 2.015 |
| Gammatone features | Guitar | 13.508 | 10.020 |
| | Drum | 253.284 | 7.146 |
| Gammatone + Middle Ear Filter | Guitar | 42.963 | 12.718 |
| | Drum | 333.045 | 17.921 |
| Gammatone + Eqloud Filters | Guitar | 27.746 | 11.335 |
| | Drum | 194.104 | 11.257 |
| Gammatone + Both auditory filters | Guitar | **48.011** | **19.806** |
| | Drum | **411.951** | **20.844** |

Table 6.1: Model performance with different preprocessing auditory filters. Experiments I used different instrument loops, and experiment II used audio generated by different audio effects.

There are two audio sets used in two experiments. Experiments I used different instrument loops, and experiment II used audio generated by different audio effects. The generation of the second audio set is described in Table 6.2. The results of Experiment I is provided in the first column of Table 6.1. Since the higher the psuedo_F index is, the better quality the clusters are supposed to be, the best performing model is the one that uses both middle ear filter and equal loudness filters in preprocessing. In most cases, using Gammatone features improves the performance compared to baseline system where MFCC features are used. This result is encouraging due to the fact that it is assumed that Gammatone features are better representations of timbre information. In addition to this,

the drum loops provide a good performance which is much better than any other cases in Experiment I. It is because the first experiment is simpler than the second one. At the same time, drum is the only percussion instrument among the test data. It has a much different timbre comparing with the others.

The second experiment is designed for the same purpose, to evaluate the preprocessing filters. The four audio effects used in this experiment are developed in SAFE project [Stables et al., 2014]. These effects are used to process mono-instrument loops taken from the AppleLoops library. There are 28 acoustic guitar loops and 4 audio effects. Compressor, equaliser, reverb, and distortion are used to generate this audio set. The details of the parameter settings of each effect are given in Table 6.2. According to the settings, $28 * 4 = 112$ audio files are generated. Their loudness is normalised to avoid its influence. The proposed models are used to estimate the similarity of every possible pairs of audio. The inter and intra class measure is then performed for the four classes using Pseudo_F index. The results are provided in the second column of Table 6.1.

| Effecs | Compressor | Equaliser | Reverb | Distortion |
|---|---|---|---|---|
| Settings | Thre: 42-45dB<br>Ratio: 13-16dB | Gain: 9dB<br>Freq: 400-700Hz | Dry/Wet:<br>10%-13% | Drive:<br>32-35dB |

Table 6.2: Audio effect parameters settings.

Due to the fact that the focus of the second experiment is on subtle timbre information, the value of pseudo_F index is much smaller than the first experiment. Meanwhile the conclusions are also consistent with the first experiment. The best performance appears when both auditory filters are used, and also Gammatone features improve the performance compared to MFCCs. In the optimised model, both filters will be applied.

### 6.3.2 Feature extraction experiments

It has been discussed in Section 6.2.2, there are three types of filter banks considered in this research: Mel-scale filters, Gammatone and Gammachirp filters, because they are better fit with data obtained from physiological studies of the human auditory system [Shao et al., 2009]. In the first experiment, the author compares three types of features derived by these three types of filters introduced in Section 6.2.2. The correlation between compressor parameters, *Ratio*, and the response of the proposed model will be calculated and used as a measure. Secondly, the model's sensitivity to ballistic parameters of DRC is also evaluated when applying delta features on top of auditory features, c.f. Figure

6.7. The delta features are considered because the attack/release time of DRC will change the transient part of the audio which can be measured through characterising temporal differences.

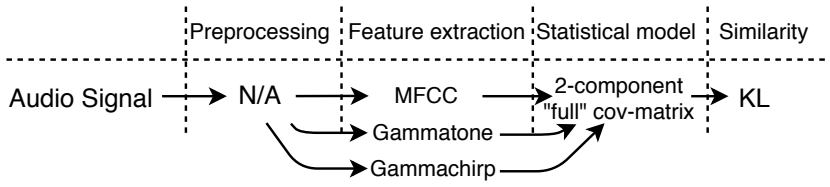



Figure 6.6: Assessment of auditory filter bank types

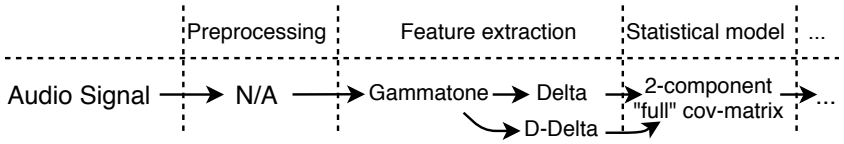


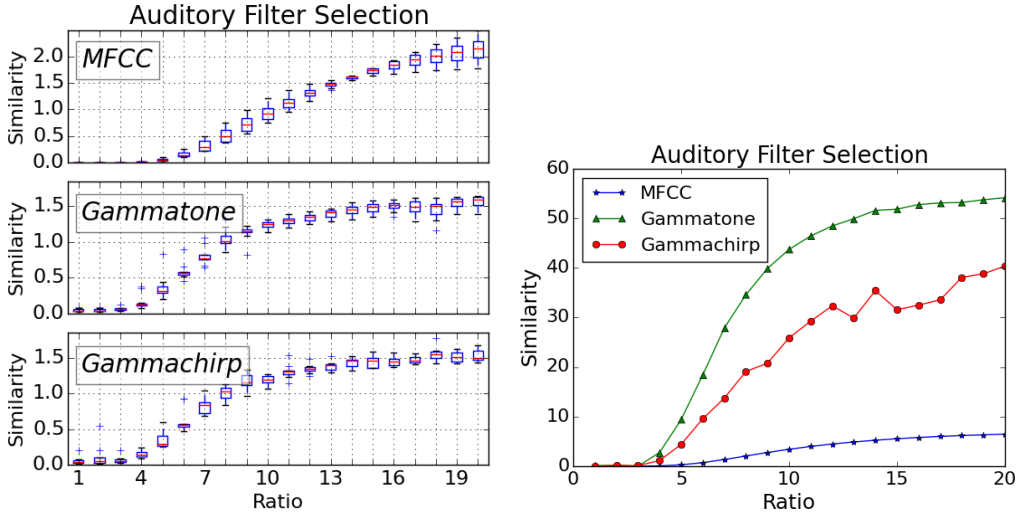

Figure 6.7: Assessment of first and second order delta features

**Auditory features**

The experiments are designed to evaluate which feature improves the model's performance. The output of the model designed which has the highest relevance and sensitivity to the DRC parameter will be applied in the final design. The isolated notes from the RWC database are compressed with increasing compression ratio. The similarity between the compressed audio and its uncompressed version is calculated. The higher the compression ratio, the greater the distance between the compressed and uncompressed sound is expected to be. The correlation between the similarity model's output and the ratio is calculated as well as the range between the lowest similarity and the highest. The former measure represents relevance while the latter indicates sensitivity. For decorrelation and to concentrate energy in the lower coefficients, DCT is used after the auditory filters in the same fashion as with the calculation of MFCCs. Implementation details are provided in Section 6.2.2. Figure 6.8 provides an example result. Figure 6.8(a) shows the changes in mean-normalised similarity for 10 violin examples. Each sound is compressed using 20 ratio settings within the range [1,20] with a step of 1. The threshold is set to -37.5dB to make sure the compression would actually take place. The rest of the parameters are kept constant for all audio files. The results indicate that the MFCC features provide nearly linear relation between ratio and similarity once the ratio parameter is above 5. However,

it reflects little of the difference when the ratio is relatively small. On the contrary, Gammatone and Gammachirp are able to reflect small ratio differences better. Figure 6.8(b) illustrates the non-normalised result of one of the violin examples. It is clear that Gammatone and MFCC features produce monotonically increasing results, but Gammachirp does not. Furthermore, MFCC features exhibit a substantially lower range of differences compared to Gammatone features. This indicates that Gammatone features are more sensitive to the change of DRC parameters.

A more general experiment has also been carried out with a larger audio set. One hundred and eighty isolated notes from the RWC database are manually compressed with different compression ratios. Sixty examples of violin, piano, and guitar samples are used respectively. Each sound is compressed using 20 ratio settings within the range [1,20] with a step of one. This process produces 1200 compressed recordings for each instrument. For every 20 audio files that are generated from the same source, we compare the predicted similarity with the ratio setting. The results of correlation and range are averaged across 1200 cases for each instrument. The recordings are between 5 to 10 seconds each. These results are presented in Table 6.3. The results show that MFCC features have the highest correlation with ratio, however, the Gammatone feature provides a much larger range, which suggests more sensitivity. The difference between MFCC and Gammatone features are not significantly different. In the comparison with the results given in Figure 6.8, the Gammatone filter appears to be the best choice among the three auditory filter models.




(a) 10 violin audio

(b) 1 violin audio

Figure 6.8: Similarity changes with the increase of compression ratio, comparing three different types of filter banks used as audio features.

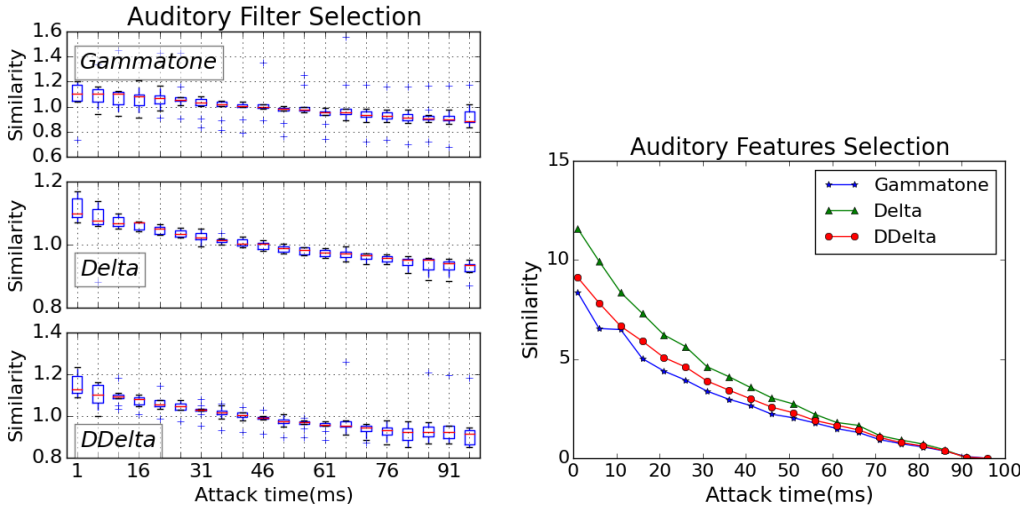
| | Correlation | | | Range | | |
|---|---|---|---|---|---|---|
| | M | T | C | M | T | C |
| Violin | **0.98** | 0.90 | 0.84 | 11.92 | **74.31** | 60.80 |
| Piano | **0.96** | 0.89 | 0.90 | 13.28 | **63.72** | 29.28 |
| Guitar | **0.98** | 0.89 | 0.89 | 23.49 | **70.81** | 41.68 |

Table 6.3: Average correlation and range between the output similarity and the compression ratio, where "M" stands for MFCC, "T" stands for Gammatone, and "C" represents Gammachirp. (a) contains normalised results and (b) shows the original scale.

**Dynamic features**

As mentioned in Section 6.2.2, attack and release times are essential temporal parameters of the DRC. Using delta features along with Gammatone features can provide a benefit when it comes to detecting the small differences, especially in the transient parts of audio signals. In the next experiment, delta and delta-delta features are compared on top of the Gammatone features, c.f. Figure 6.7. The feature vectors are shown in Eqn.6.4 and Eqn.6.5. For simplicity, $k$ is chosen to be 1 for both equations. The attack time of a compressor is varied and applied to violin notes. The similarity between the processed audio and its uncompressed original is compared. The results are shown in Figure 6.9. Figure 6.9(a) shows the difference when using only Gammatone features, Gammatone features with first order delta features, and with second order delta features. The results are mean normalised. Figure 6.9(b) illustrates a single non-normalised curve of one violin note. It is clear that the Gammatone feature plus the first order delta features has the largest range, and the curve is the smoothest among the three feature combinations. We also extend this experiment to isolated notes of piano and guitar. We compressed the note using different attack times within the range [1,100]ms with a step of 5ms. The average correlation and range for three types of instruments are given in Table 6.4, where "T" represents Gammatone features only, and "D", "DD" represents the Gammatone features plus the first and the second order delta features. Due to the negative correlation between the similarity and attack time, we display the absolute value of the average correlation. Gammatone plus the first order delta features provide the highest correlation for all three instruments. The second order delta features have a lager range in two out of three cases. Balancing the results of both correlation and range, Gammatone with the first order delta features is an appropriate choice. The attempt of applying S-G filter introduced in Section 6.2.2 has also carried out to smooth the delta features. The results have not significantly

improved (less than 3% in most cases), therefore this step is skipped in the final design to avoid introducing additional computational expense for a small uncertain benefit.




(a) 10 violin audio

(b) 1 violin audio

Figure 6.9: Similarity changes with increasing compression attack time, comparison of cases with and without using delta features

| | Correlation | | | Range | | |
|---|---|---|---|---|---|---|
| | T | D | DD | T | D | DD |
| Violin | 0.84 | **0.87** | 0.84 | 9.51 | 11.50 | **13.20** |
| Piano | 0.87 | **0.88** | 0.76 | 1.63 | **2.47** | 2.37 |
| Guitar | 0.52 | **0.60** | 0.53 | 12.02 | 19.02 | **20.62** |

Table 6.4: Average correlation and range between the output similarity and the compression attack time, where "T" represents Gammatone features, "D" represents Gammatone features plus first order delta feature, and "DD" represents Gammatone features plus second order delta feature.

### 6.3.3 Statistical modelling

The selected features are subsequently modelled using a GMM. As per Figure 6.3, two to three Gaussian components would be sufficient based on empirical observation. An exhaustive search has been carried out to confirm this assumption. The model is tested using Gaussian components ranging between one to four, and also test the type of covariance matrices including 'spherical', 'tied', 'diagonal' and 'full' [Bishop, 2006]. The performance of GMMs is measured using Bayesian information-theoretic criteria (BIC) [Schwarz, 1978]. For each audio, the model that yields the lowest BIC will be selected. This search is conducted using mono-timbral loops from AppleLoops, more specifically, 40 examples of each instrument (drum, violin, accordion, piano, guitar, and bass) as shown in Table 6.5. Since

1 or 4 Gaussian components and 'spherical' or 'tied' covariance matrices are not selected in any cases, Table 6.5 only shows the results for successful selected models. Using the full covariance matrix yields the best performance in most cases, which suggests that the features are not fully decorrelated. The most common choice for Gaussian components is 3 (77.5% in average). Therefore, the final statistical model we will use is a GMM with 3 components. Each component has a full covariance matrix.

| Drum | Components | | Var | | Piano | Component | | Var | |
|---|---|---|---|---|---|---|---|---|---|
| | 2 | 3 | diag | full | | 2 | 3 | diag | full |
| | 30% | **70%** | 0 | **100%** | | 10% | **90%** | 0 | **100%** |
| Violin | Components | | Var | | Guitar | Components | | Var | |
| | 2 | 3 | diag | full | | 2 | 3 | diag | full |
| | 42.5% | **57.5%** | 0 | **100%** | | 15% | **85%** | 0 | **100%** |
| Accordion | Components | | Var | | Bass | Components | | Var | |
| | 2 | 3 | diag | full | | 2 | 3 | diag | full |
| | 25% | **75%** | 20% | **80%** | | 12.5% | **87.5%** | 0 | **100%** |

Table 6.5: Gaussian components and variance matrix selection for GMM. The model is tested using several Gaussian components and the type of covariance matrices. The performance of GMMs is measured using Bayesian information-theoretic criteria (BIC). For each audio, the model that yields the lowest BIC is selected. The selection rate is provided in this table.

In conclusion, the experiments above have been conducted to select the design of each component in this similarity model. The selection criteria aimed to make the model more sensitive to the DRC's parameters, as well as timbre and loudness, the two audio perceptual aspects of sound that are usually affected by the DRC.

Based on the experiments, the model is designed as follows. In the preprocessing stage, the raw audio is being processed using both middle ear filter and equal loudness filters, and 40-frequency bands Gammatone filter bank ranging from 0 to 22050Hz are used for feature extraction. DCT is applied to the log energy of windowed subband outputs of the Gammatone filter bank. The top 13 coefficients are selected as audio features. First order delta features are computed from the features and a stacked feature vector is then formed. The statistical model representing the features is a 3 components GMM with full covariance matrices. The KL divergence approximated with variational Bayes [Hershey and Olsen, 2007] is then used to estimate the similarity of the GMM models. Figure 6.10 indicates the final computation workflow. This model will be used in the overall performance evaluation detailed in the next section.

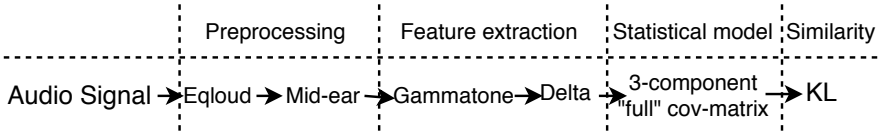



Figure 6.10: Final system components and workflow

## 6.4 Evaluation

After finalised the components of the model, the proposed model design need to be tested to determine whether the model corresponds well enough to this research's very specific task. This means to evaluate the similarity that is caused by DRC. In this section, an overall evaluation of the similarity model will be carried out. There are two experiments. The first one tests the model performance when changing the compression parameters. The second evaluates the relation between the similarity model outputs and loudness differences.

### 6.4.1 Dynamic Range Compression Parameters

In this experiment, the model's performance is evaluated against DRC parameters. The focus of this Thesis is on the following four DRC parameters: threshold ($\theta$), ratio ($\gamma$), attack time ($\tau_a$) and release time ($\tau_r$). As discussed in Section 2.1.3, there are other common compressor parameters that are not considered in this work, including knee type and make-up gain. Knee type controls the smoothness of gain reduction around the threshold and typically results only in a subtle perceptual effect. Make-up or output gain is purely a boost of level, therefore these two parameters are excluded in the test. As in previous experiments, the compressed notes are compared with their uncompressed versions. Sixty isolated violin notes from RWC database are used. The normalised results are shown in Fig. 6.11. The parameters $\rho = \{\theta, \gamma, \tau_a, \tau_r\}$, i.e. threshold, ratio, attack time, and release time, are varied as follows: $\theta \in [10, 12, ..., 46, 48]dB, \gamma \in [1, 2, ..., 19, 20], \tau_a \in [5, 10, ..., 95, 100]ms$, and $\tau_r \in [10, 60, ..., 910, 960]ms$. The curves in Figure 6.11 show that the divergences reflect the changes in the compressor for all the parameters. Since the threshold and ratio are in dB scale, their similarity is not changing in a linear fashion. Attack time and release time are linear in the time domain, and consequently the similarity curve shows a change that is closer to linear. A smaller attack time changes timbre more significantly than larger attack, therefore the correlation between the similarity and the attack time is negative, while the opposite is true for release time.

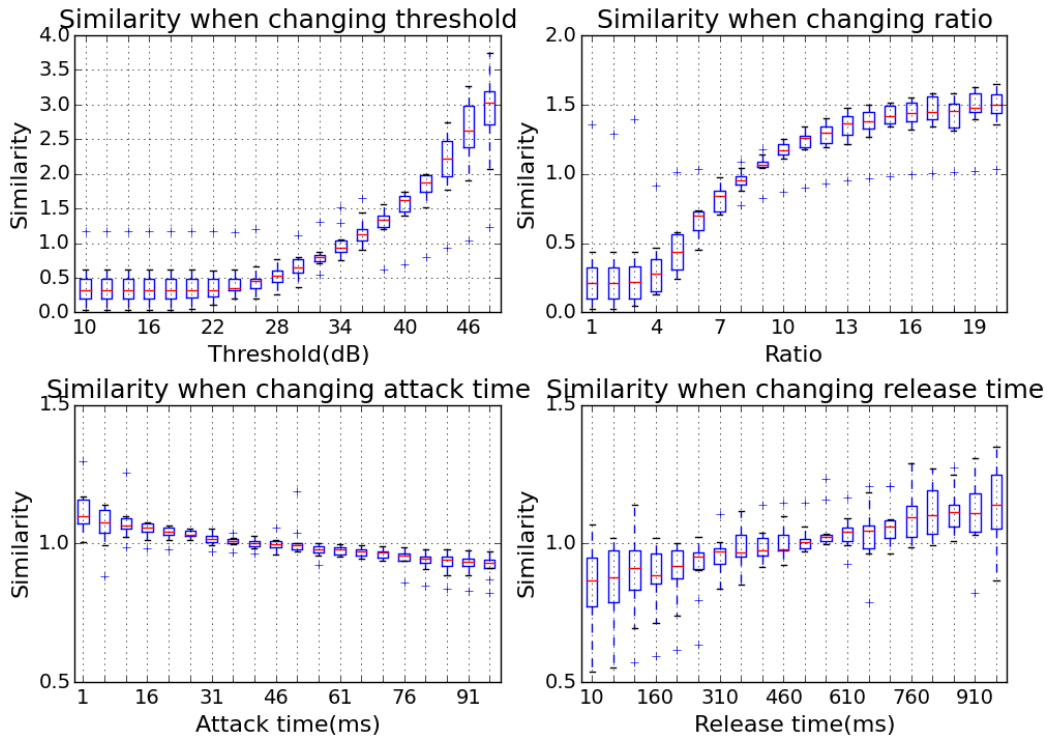



Figure 6.11: Similarity change with the increase of compression parameters for threshold, ratio, attack and release time.

### 6.4.2 The Influence of Loudness

The question investigated in the following assessment is whether the similarity model reflects changes in loudness, and whether the relationship between the two is linear. To answer the first question an experiment is conducted on 60 violin loops for testing and one violin loop as reference from AppleLoops. The similarity between the reference and test loop are compared with their loudness difference. The integrated loudness is calculated using the EBU-R 128 algorithm [ITU]. The loudness curve and the similarity curve show a modest Pearson product-moment correlation (R=0.446). This indicates that the system is able to account for similarity in loudness, i.e., the similarity estimation is related to loudness change significantly, but it also shows that if the primary interest is in timbre, the samples should be loudness normalised before similarity estimation.

To investigate further how loudness is related to the proposed model, another experiment is carried out under a more controlled situation. Sixty drum loops are selected from AppleLoops. Their integrated loudness[1], introduced in EBU R128, is modified. The loud-

[1]The loudness level normalisation tool in Hindenburg Journalist PRO is used, a multitrack audio editor. https://hindenburg.com/products/features/loudness-meter.

| Relation with *loudness* | *simi* | $\sqrt{simi}$ | $log(simi)$ |
|---|---|---|---|
| error | 0.071 | **0.010** | 0.037 |
| corelation | 0.971 | **0.999** | 0.977 |

Table 6.6: Relations between similarity and loudness using different transformation of similarity.

ness normalisation is applied for each loop in 7 levels (0dB, -3dB, -6dB, -9dB, -12dB, -15dB, and -18dB) starting from 23dB. For each loop, the other 6 levels are compared to 0dB level and a 1∗6 vector is generated, i.e. $simi = [s_1, ..., s_6], s_i = similarity(drum_{0dB}, drum_{i*3dB})$, $i * 3$ regarding to the 7 levels. A linear relationship is assumed between loudness and similarity, i.e. the relation between $loudness = [3, 6, 9, 12, 15, 18]$ and *simi* is linear. The vector *simi* is fitted to a linear curve line using the least squares method, $\hat{simi} = a \times loudness + b$, and calculate the fitting mean absolute error. For easier comparison, the *simi* is normalised to the range $[0, 1]$. The average error given 60 loops is detailed in Table 6.6. The average correlation between *simi* and *loudness* is provided as well. The analysis is extended by taking the square-root and log of *simi* and repeating the same process. The similarity itself shows a strong correlation with loudness. A stronger correlation and less error is produced when comparing with the square of similarity as it is shown in Table 6.6.

## 6.5 Conclusion

This chapter discusses the design of an audio similarity model which targets timbre and loudness.

The commonly used audio similarity estimation methods are outlined in Section 6.1. The major difference between the proposed model compared to other methods is that the proposed design targets subtle perceptual aspects of sound affected by audio effects. The proposed model is designed in four stages. The specific methods and features that are used in each stage are decided by the results of specific experiments discussed in Section 6.3. The proposed final model is illustrated in Figure 6.10. The overall performance is outlined in Section 6.4 focussing on tasks relevant to audio effect parameter estimation and classification. The results show that the proposed system is sensitive to loudness and timbre. This model shows sensitivity to the changes in sound introduced by varying DRC parameters, although to different extents.

The system uses middle ear filter and equal loudness curves as preprocessing which

can be regarded as related to the key perspective aspects, loudness and timbre. There are other aspects of auditory models that need to be considered if the targeted perceptual aspects are changed. The results indicate that using preprocessing (i.e. using the middle ear and equal loudness filters) does not always guarantee a better performance in the tasks examined. In theory, and most of the test cases, auditory filters allows the system to better reflect the changes in perceptual attributes induced by effects.

The proposed preprocessing filters may therefore be traded off if computational cost is of concern. It is also worth considering how the given task relates to the experiments presented here which could be a helpful hint suggesting when additional experimentation may be beneficial. The system also applies a complex parametric statistical model to represent the generated audio features. A three components GMM is chosen in the design. Using three components is a good trade off as singularity (i.e. zero probability mass assigned to a component during the EM parameter estimation process) is easier to occur with more Gaussian components. Therefore having less Gaussian models is also more likely to avoid the problem with calculating KL divergence.

The system can be applied to the proposed intelligent control system for dynamic range compressor [Sheng and Fazekas, 2017, 2018a,b]. This system aims at using specifically designed audio features and a regression model to bring the input audio sounds as close as possible to a reference audio. The similarity model can be used as an evaluation tool to compare the output audio of the system and the targeting audio. It can also serve as an optimisation function to introduce perceptual aspects into model design or hyper-parameter selection of intelligent control tools for audio effects. This can bypass conducting time consuming and potentially expensive listening tests at various interim stages of the development.

# Chapter 7

# Optimisation and Subjective Evaluation of the Intelligent Control System

In the previous chapter, the author introduced an audio similarity model targeting loudness and timbre. This approach has demonstrated its sensitivity for the changes of compressor parameters. In the first part of this chapter, the author introduces a scheme that uses the similarity model as an optimisation of the intelligent control system. The second half of the chapter includes a subjective listening test. The primary aim of the listening test is to find the minimal audible difference for the ballistic parameters of the DRC. Since the ballistic parameters are the ones that are harder to predict, this subjective test is able to access whether the prediction model has reached a level of performance that makes it suitable for applications in real-world music production. If the error rate reaches lower than the audible threshold, it can be considered as optimal prediction. The first experiment in this chapter is an optimisation of the prediction model with an emphasis of perceptual aspects. The second experiment is a perceptual test to support the prediction model. The results from both experiments can serve as perceptual support for the designed intelligent model.

## 7.1 Optimisation Design

The prime motivation of designing a similarity model is to evaluate the performance of the intelligent system along its development. The author also considers to use it as an optimisation method to improve the performance of the intelligent system. The adaptation of a multi-stage search algorithm has been considered. The intelligent system will predict the parameter settings at the first stage, and the similarity model will be used for a *local search* at the second stage. Details of the design is presented in Section 7.1.2. The results and evaluation follow Section 7.1.3.

### 7.1.1 Motivation

To recap, the training process of the intelligent system, c.f. Section 3.2.2, used the audio pairs such that one is the original and the other is a compressed version. Therefore, the regression model can be trained using the ground truth of the compression parameters. The system is then evaluated using the audio perceptual similarity model. This design shows decent performance in many scenarios that have been tested in the previous research. A clear trend can be observed that when applying the system to real world scenario, i.e. when the reference audio and the input audio are not from the same origin, c.f. Section 3.2.2 the proposed system still provide decent performance. In this situation, there is no ground truth for each pair of audio.

This section is aiming at combining the two stages. It is reasonable to insert the similarity model into the intelligent system as an extra optimisation stage. This approach would incorporate the similarity measurement into the prediction process. The overall system thus takes audio perceptual information into account, which is what this research is aiming to achieve.

There are several applications in sound synthesis research that applies parameter search strategies. Mitchell and Creasey [2007] demonstrate several Evolution Strategy-based algorithms to perform synthesis matching, which equates to finding the synthesising parameters that will provide a match to the target audio. The parameters are selected by the best performing algorithm. The Subsequent work, [Mitchell, 2012], discussed the optimisation issue more deeply. This paper also provides a novel clustering evolution strategy. Authors in [Gabrielli et al., 2018] outline a multi-stage optimisation strategy. This work combines two stages such that the first one contains several neural networks for parameter predic-

tion as a global search. The second stage runs a local search after the first stage using a Random Iterative Search (RIS). Inspired by these works, the author designed a two stages optimisation strategy to predict DRC parameters.

### 7.1.2 Method and dataset

Since the performance for the simple situation is already good with relatively small error rates that are probably inaudible, this research is aiming at improving the performance for the complex scenario, i.e. predicting DRC parameters for polyphonic audio tracks. At the end of Chapter 5, an experiment conducted on the polyphonic track from MedleyDB dataset [Bittner et al., 2014] is presented. The optimisation is built as a further development of this experiment. The data used in this research is the drum loops dataset described in Table 5.11. It also includes 50 audio excerpts from MedleyDB, which are compressed in the same fashion as Table 5.11. With less data, the resolution of parameter steps can be slightly finer. Details are given in Table 7.1. There are 5 settings for each parameter of $\boldsymbol{\rho} = \{\theta, \gamma, \tau_a, \tau_r\}$, and each audio is compressed using the parameters given by the permutations. In a real world scenario, the input audio and the reference audio are from different origins. This scenario makes the direct numerical evaluation impossible. The similarity model designed in Chapter 6 becomes convenient for this scenario. The two stage optimisation algorithm is proposed in this chapter. Based on the previous research, the siamese model is used for feature embedding calculation, and a trained random forest regressor is used for the first stage prediction. This process serves as the global search stage. The similarity model is used in the second stage as a local search step. A Random Iteration Search using parameter perturbation is applied in this stage. The final predicted parameters are then used to generate the output audio. The system diagram is illustrated in Figure 7.1.

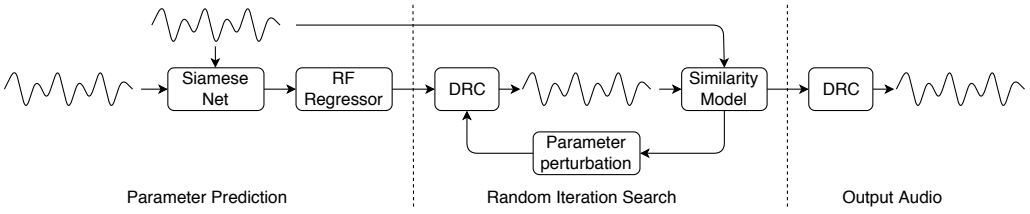



Figure 7.1: Two-stage algorithm overview

As it is presented in the system diagram, the CNN-based siamese model and random

| | dataset generation | dataset size |
|---|---|---|
| D4P | $\theta$: 1 to 50dB with step of 1dB<br>$\gamma$: 1.38 to 20 with step of 0.38<br>$\tau_a$: 1 to 99ms with step of 2ms<br>$\tau_r$:10 to 990ms with step of 20ms | mixed audio excerpts: 50*625 |

Table 7.1: Dataset details for data generated when changing four parameters together

forest regression model is pre-trained by the experiments in Chapter 5. The siamese model that predicts 4 parameters jointly is trained by dataset described in Table 5.11 and 7.1. The CNN-based siamese model structure follows Table 5.1. The random forest model is trained on the learnt feature embeddings as well as the ground truth compression parameters. The experiment procedure is shown in Figure 7.1. The input audio and the reference audio are randomly selected pairs from the original audio and the compressed audio. The first step of prediction can provide a set of parameters, and the interim audio can be generated based on this. The second step can be referred to as a local search. The similarity between the interim audio and the reference is used as a guideline for the random iteration search. The details of this strategy is provided in the pseudo code Procedure 5.

**Procedure 5**

**Input:**

$n$: test data set size;
$\boldsymbol{O}$: a list of uncompressed audio files, with the size of $n$;
$\boldsymbol{R}$: a list of compressed audio files with the size of $n$, will be used as references;
$\boldsymbol{P}$: the predicted parameters out of $\boldsymbol{O}$ and $\boldsymbol{R}$;
$lr$: learning_rate, the speed of updating the parameters;
$m$: the iterations that this algorithm is going to run;

**Output:**

$\boldsymbol{P\_new}$;

**for** $i \in range(n)$ **do**
  $a$ = DRC(input = $O_i$, para = $P_i$)
  $s$ = Similarity_Model($a$, $R_i$)
  $\hat{s} = s$
  $b = P_i$
  **for** _ $\in range(m)$ **do**
    $\Delta = \hat{s} * (b * \Gamma) * \Lambda$
    $\hat{P}\_new_i = lr * \Delta + b$
    $\hat{a}$ = DRC(input = $O_i$, para = $\hat{P}\_new_i$)
    $s_t$ = Similarity_Model($\hat{a}$, $R_i$)
    **if** $s_t < \hat{s}$ **then**
      $b = P\_new_i$
      $\hat{s} = s_t$
    **end if**
  **end for**
  $P\_new_i = b$
**end for**

The random algorithm is based on the random perturbation of the parameter space.

This algorithm has two audio sets as input. $\boldsymbol{O} = \{O_1, ..., O_n\}$ is the input audio set with the size of $n$, and $\boldsymbol{R} = \{R_1, ..., R_n\}$ is the reference audio set. The parameter out of the first stage, i.e. the prediction model is presented as $\boldsymbol{P} = \{P_1, ..., P_n\}$, with the size of $[n, 4]$. The algorithm has a learning rate, $lr$, which is smaller than 1, and the number of iterations, $m$. For each audio pair $O_i$ and $R_i$, the initial similarity $s$ is calculated and the predicted parameters $P_i$ is assigned as the temporal best parameters, i.e. $b$. The random perturbation of the parameter space is performed $m$ times. In line 7, $\Gamma$ is a sparse random vector with value $\in$ 0,1. Using this vector allows only a random subset of the parameters to be perturbed at each iteration. $\Lambda$ is a normally distributed random vector. We use a Gaussian distribution with the mean of 0 and variance of 1. $b$ is the best performing parameter setting that appeared in the previous iterations, and $\hat{s}$ is the similarity between reference audio and the compressed audio using the current best performing parameters. This perturbation is then multiplied by a learning rate and added to the best parameters so far. A new compressed audio $\hat{a}$ is generated according to $P\_new_i$. The new similarity $s_t$ is calculated. If new similarity is smaller than $\hat{s}$, the best parameters are updated to $P\_new_i$, and the best similarity is updated to $s_t$. After $m$ iterations, the best parameters are assigned to $P\_new_i$ as output.

### 7.1.3 Evaluation

The evaluation of the optimisation design is discussed in this section. For drum loops, the reference audio is randomly selected from the 40000 compressed audio excerpts. The input audio is randomly selected from the 64 original audio. Similarly, for polyphonic music, the reference audio is randomly selected from 31250 compressed audio excerpts. The input audio are randomly selected from the 50 original audio. The similarity results from the reference and input audio pairs, after the first stage of the prediction model, and after the second stage of the random iteration search are provided in this section. We set the number of iterations to 20 and the learning rate to 0.1. This experiment generates 1000 audio pairs, and the similarity after two stages is represented in Table 7.2.

| | Initial | First stage | Second stage |
|---|---|---|---|
| Average similarity - Drum | 308.221 | 105.087 | 82.549 |
| Average similarity - Poly | 263.088 | 248.071 | 171.590 |

Table 7.2: Similarity properties after two stages

The second stage successfully decreases the distance between the reference and the output. If we normalise the result by setting the initial similarity to 1, for drum loops the distance drops to 0.268, and 0.652 for the polyphonic music. For reference, a similarity evaluation has been conducted on the random audio pairs for both drum loops and polyphonic music without optimisation. The evaluation used both handcrafted features and learnt feature embeddings. The average results are presented in Table 7.3. The optimisation method is able to provide a closer perceptual distance.

| | | Average Similarity |
|---|---|---|
| Drum | Handcrafted features | 287.878 |
| | Feature embeddings | **105.087** |
| Poly | Handcrafted features | 257.563 |
| | Feature embeddings | **248.071** |

Table 7.3: Similarity evaluation in a real world scenario on drum loops and polyphonic music, and predicting four parameters simultaneously.

To have a better understanding of the performance, other related features are also extracted for the output audio of the two stages. The crest factor can be used to indicate the dynamic range of the audio signals. This feature can not represent all the aspects of sounds that the DRC can change, but to a certain extent, it can be used to measure the performance of the system. If the crest factor differences are getting closer, it implies the system is able to make the audio similar in terms of dynamic range. Loudness is also measured for similar purpose. Loudness is computed using Steven's power law [MacKay, 1963]. This algorithm computes loudness as the energy of the signal raised to the power of 0.67. The results are presented in Table 7.4 and 7.5, the data are the difference between the output and the reference audio. The two features show a similar trend as the similarity measure. The second stage managed to reduce the distance for these features as well. The algorithm has thus provided improved performance.

| | Initial | First stage | Second stage |
|---|---|---|---|
| Average crest factor difference - Drum | 70.051 | 68.270 | 56.321 |
| Average crest factor difference - Poly | 335.568 | 331.921 | 232.465 |

Table 7.4: Crest factor difference after two stages. "Initial" represents the initial difference between the input and reference, and "First stage & Second stage" are the differences between the output and reference.

| | Initial | First stage | Second stage |
|---|---|---|---|
| Average loudness difference - Drum | 12.273 | 11.261 | 10.953 |
| Average loudness difference - Poly | 63.717 | 60.544 | 45.801 |

Table 7.5: Loudness difference after two stages. "Initial" represents the initial difference between the input and reference, and "First stage & Second stage" are the differences between the output and reference.

## 7.2 Subjective evaluation on the audibility threshold of DRC parameters

The research up to this subsection has been aiming at pushing the boundary of the prediction model. To understand the efficiency of the model in real-world conditions, i.e., compare to human performance in a similar task, this experiment is designed to discover the minimum audible difference between the ballistic parameters i.e., release and attack times of the DRC. The test and the analysis is presented in this subsection, starting with the motivation of the test. The procedures are outlined is Section 7.2.2. Evaluation and analyses are followed in Section 7.2.3. The conclusion of this experiment along with the conclusion from the optimisation experiment in the first half of this chapter will be presented in Section 7.3.

### 7.2.1 Motivation

As discussed several times before, DRC is one of the most commonly used audio effects in music production. This Thesis is aiming at designing an intelligent computational system for predicting the parameters of DRC. The performance can be evaluated using the computational model proposed in Chapter 6. A more convincing evaluation would be a subjective test. The results from this experiment can inform the efficiency of the computational system because we can safely assume that the prediction errors that are smaller than the audibility threshold can be considered inaudible. Therefore this experiment is able to serve as a support for the numerical evaluation results presented in the series of previous chapters.

This listening test is designed to focus on the ballistics parameters partly due to computational reasons. It is also because these parameters turned out harder to predict and there are already studies focussing on the perception of loudness differences which allow some insight into the audibility of compression ratio and threshold. To study the audibility

threshold ideally requires a large audio dataset. Using all four parameters, the audio sets would be much larger, therefore the experiment would be difficult to design and complete in reasonable time while also avoiding the effects of listener fatigue. Since the effect of threshold and ratio is more profound and has been discussed more often, the author focuses on attack and release times in this test. There has been some discussion on the perceptual impact of the ballistics parameters of the DRC in the literature, e.g. the influence of these parameters to the perception music style is discussed in [Bromham et al., 2018]. A broader discussion on DRC's impact on music has been presented in [Wagenaars et al., 1986]. Apart from the musical aspects, the audio perceptual impacts of the DRC is analysed along with other audio effects in [Wilmering et al., 2013]. Little discussion has appeared in the literature about the audible difference of the DRC parameter changes. This experiment can be the first step in a series of more detailed experiments. The experiments presented here also aim to provide a better understanding of the perceptual behaviour of the DRC in general. There are further possibilities of discovering the relations between age, experience, and the perception of the DRC using the data collected in the study.

As introduced earlier, the main purpose of this test is to detect the minimum audible difference of DRC ballistic parameters. We assume that when an audio is compressed with a ballistic parameter set to $t$ and $t + \delta$, the difference between the two compressed audio is not audible even to experienced ears, if $\delta$ is small enough. This study is aiming at finding the threshold $\delta$.

### 7.2.2 Experiment design

Audio effects can alter many perceptual aspects of the audio [Wilmering et al., 2013]. The amount of perceived change induced by an audio effect may also depend on the audio material, e.g. the type of instrument sound the effect is operating on. Taking this into account, four types of mono-instrument tracks were used as test materials: Bass, Guitar, Drum and Vocal. All audio tracks are 5-6 seconds long with a normalised loudness of -23dB. The four tracks are compressed using the parameter settings in Table 7.6. The tracks are compressed in the way that we kept three parameters fixed and changed one parameter with 5 different settings for each parameters. The fixed parameters have two modes, a light and a deep threshold with a fixed ratio. In this way, the test is conducted on 4 instruments tracks, 2 compression modes, 2 ballistic parameters, and 5 settings of each parameter, therefore 4*2*2*5=80 compressed audio tracks are generated. The attack time

and release time are set to 10ms-30ms and 100ms-300ms respectively because they are the most commonly used ranges based on the informal interviews of professional producers.

| | Parameter settings | Mode |
|---|---|---|
| Attack time | 10ms-30ms, step of 5ms | *Light*: threshold: -35dB, ratio: 1:5<br>*Deep*: threshold: -45dB, ratio: 1:5 |
| Release time | 100ms-300ms, step of 50ms | |

Table 7.6: DRC parameter settings: 2 modes of which one is heavy compression and the other is light; 5 settings for attack time and release time.

An ABX listening test is conducted for this study [Clark, 1982]. The listeners are presented a reference audio and two candidate audio excerpts. The reference audio would be the same as one of the candidate audio, and the other is the same instrument track with same compression mode but a different setting. The listeners are asked to pick the one which is different from the reference. For each mode of attack time, we compare 30ms with 25ms, 20ms, 15ms and 10ms. The same procedure applies to release time. Therefore, for the each instrument (4), each mode (2) and each parameter (2) we can generate 4 pairs. The listeners are asked to compare 4*2*2*4=64 pairs. Having a reference helps the participants to focus on the different parts between the audio. It can also be assumed that if the audio pair's difference is not audible, the responses will have a random distribution. Finding the separating spot between random and not random distribution will enable determining the audibility threshold.

Sixteen participants were recruited. Since this test is highly specific, only people who have experience with production are recruited. A survey was also conducted using a post-experiment questionnaire before the listening test to collect the demographic information such as age, experience level, gender, occupations and so on. An overview of the participants is provided in Table 7.7. There is a majority of male participants. This echoes the gender imbalance of the music production industry. The distribution of proficiency and age is fairly even. Section 7.2.3 contains both global analysis and analysis with data break-down based on these information.

### 7.2.3 Analysis

Two analysis stages are presented in this section. First of all, the analysis of accuracy is performed on each group of participants. Second, the analysis is carried out for the prime purpose of this test, which is determining the audibility threshold of the ballistic parameters.

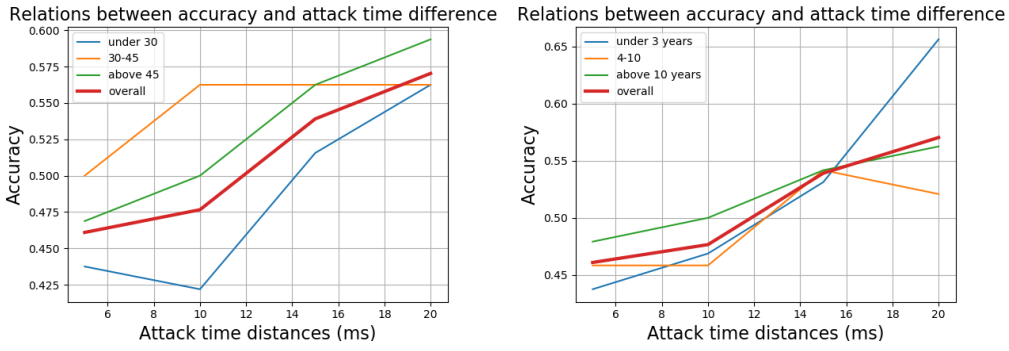
| **Gender** | | **Occupation** | | **Hearing Impairment** | |
|---|---|---|---|---|---|
| Male | 14 | Music Producer | 6 | No | 13 |
| Female | 2 | Others | 10 | Unknown | 3 |
| **Level** | | **Years of experience** | | **Age** | |
| Little | 3 | under 1 | 2 | under 30 | 8 |
| Average | 2 | 1-3 | 2 | 30-45 | 4 |
| Above-average | 6 | 4-10 | 6 | above 45 | 4 |
| Expert | 5 | 11-20 | 2 | | |
| | | above 20 | 4 | | |

Table 7.7: Overview of the participants' information.

### Individual group accuracy analysis

The first analysis is to break down the data into age groups and years the participants have spent in music production. The results are presented in Figure 7.2. The accuracy for each individual is calculated using Eqn. 7.1, where $R_i = 1$ when the response of pair $i$ is correct, and $R_i = 0$ otherwise. $T$ is the total audio pairs presented to each participants, i.e. $T = 64$. The average accuracy of the individuals in each category is displayed in the following figures.

$$Accuracy = \frac{\sum_{i=1}^{T}(R_i)}{T} \tag{7.1}$$

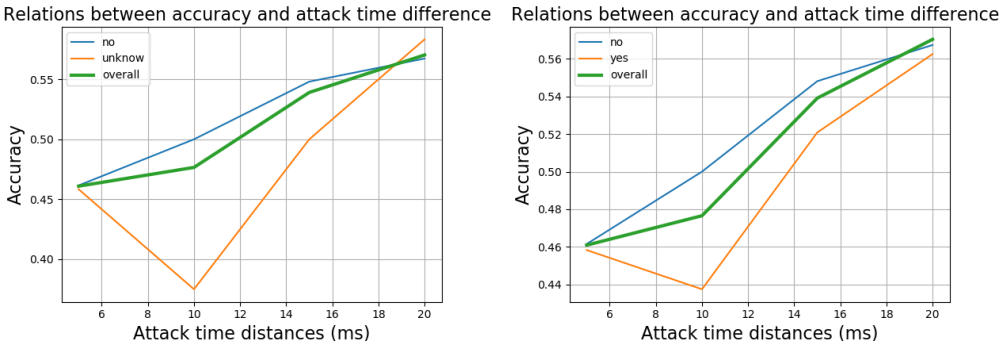



(a) age (b) years of experience

Figure 7.2: Accuracy analysis on the attack time distance between the audio pairs with respect to age and years of experience.

The assumption of the results is that the accuracy should increase with the growth of the attack time distance between the audio pairs. Most of the participants fit this assumption. Interestingly, participants with age under 30 or over 45 years fit this assumption the best,

c.f. Figure 7.2(a). This may be the effect of relatively small sample size when participants are grouped. Participants over 45 years old perform higher than average in all cases. There are 4 participants in this category, and all of them have over 20 years experience, and they are all professional music producers. This corroborates another assumption that expert producers would be more sensitive to these differences. However, the results from Figure 7.2(b) and 7.3(b) does not fully support this assumption. Although people with longer experiences in production provide a better performance when the distance are small and subtle, people with less experience show a much higher accuracy when the distance is much obvious. People with less than 3 years of experience are largely under 30 years old (75%). One possible reason causing this result is because the sensitivity of the human auditory system is better with a younger age. In other words, this result also shows that even with little training, ordinary listeners are still able to hear the changes between different attack and release time settings of the DRC. Figure 7.3(a) also indicates that there is a performance gap between the people who answer "No" to the hearing impairment and the ones who answer "Unknown", especially when the distance is small. The behaviour difference between these two groups is relatively small when the distance is large. Further experiments may be needed to confirm the impact of hearing impairment.




(a) hearing impairment

(b) occupation

Figure 7.3: Accuracy analysis on the attack time distance between the audio pairs with respect to hearing impairment and occupation.

### Audibility threshold

The main purpose of this experiment is to find out the audibility threshold of the ballistic parameters of the DRC. For each instrument, each compression mode and each parameter, there are 16 responses from participants. In this analysis, the accuracy and the statistical significance test results are represented. The chi-squared test is performed between the

uniform distribution, i.e. random distribution and the participants' responses, since the responses are binary categorical data. For the p-value lower than the alpha value, it can be assumed that the response distribution has a significant difference from the uniform distribution. The alpha value is set to 0.1 due to the fact that the problem is rather complex, therefore choosing a relaxed significance threshold is appropriate. If the accuracy is higher than 50% and the distribution is different from uniform distribution, it can be inferred that this audio pair's difference is significant. If so, the parameter distance between the audio pair can be considered audible.

One example is presented in Figure 7.4. These are the results for the Bass samples with a light compression mode and different attack times. The accuracy results show most of the responses have a better outcome than random, but only the audio pair 10ms/30ms and 15ms/30ms shows statistically significant difference compared to the the random response. Therefore, when the attack time difference is $30 - 15 = 15ms$, it is audible. While when it is $30 - 20 = 10ms$, it is not. Furthermore, the attack time audibility threshold for this case should be between 10ms to 15ms. The same evaluation is done on all audio examples. The consistency is considered as well. For example, if the accuracy of the audio pair with 15ms distance is above threshold and statistically significant, but it is not for the pair with 20ms, it can be assumed that the results of 15ms is a singular result. The audibility threshold is larger than 20ms. This is denoted in Table 7.8 as ">20ms" for attack time and ">200ms" for release time. The summary of each instrument and each compression mode is provided in Table 7.8.

| Compression Mode | Parameters | Bass | Guitar | Drum | Vocal |
|---|---|---|---|---|---|
| Light | Attack | 10-15 | >20 | >20 | >20 |
| | Release | 150-200 | >200 | >200 | >200 |
| Heavy | Attack | 10-15 | 15-20 | >20 | >20 |
| | Release | 150-200 | 150-200 | 100-150 | >200 |

Table 7.8: Audible threshold in milliseconds for 4 instruments, 2 compression modes and 2 parameters.

An important trend that can be observed in Table 7.8 is that DRC has different effects on different instruments. The audibility threshold of bass is smaller than all the other instruments. Meanwhile, the audibility threshold of vocal is much higher. It has been assumed that audio tracks with more transients should produce a smaller audibility threshold. The results corroborate with this assumption to a certain extent but not entirely. Vocal

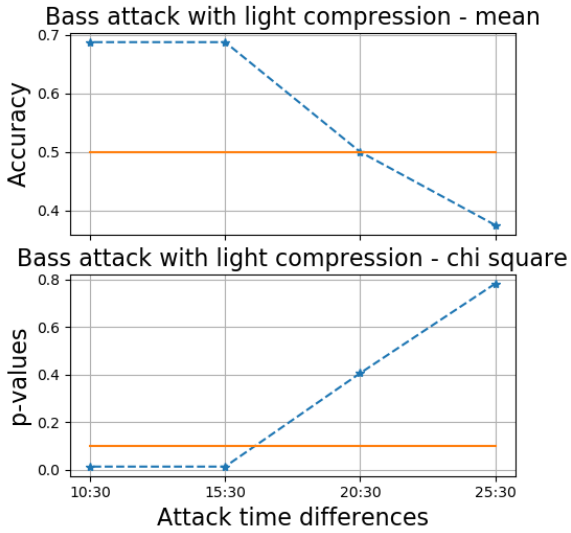



Figure 7.4: Accuracy and chi-square test analysis for attack time with Bass audio samples with a light compression mode.

signals are often dominated by steady state sounds. i.e., we can assume fewer transients in general, but vocals also include plosive and fricative sounds which are transient like, and these may be sufficient for people to identify the effects of the compressor ballistics. If this assumption holds true, the audibility threshold of drum should be small, which does not show in the results. Most of the audibility threshold of drum is over the maximum difference set in the experiment. This observation requests further investigation. The accuracies for drum are reasonable, but not consistent. For example, the accuracy for drum pair with heavy compression level with distance of 5ms is 0.81. However, the inconsistency makes the threshold much larger. This may need further experiments that focus on drum tracks to provide a more robust explanation. From the results of bass, guitar and drum, it can be implied that the audibility threshold of ballistic parameters can be smaller when the compression level is heavier. This outcome is more obvious for release time.

One of the aim of designing the listening test is that the audibility threshold can be used to inform the efficiency of the prediction models introduced earlier as the main contribution of this Thesis. The smallest prediction error for guitar and drum loops when predicting attack time is between 0.9 - 1.2ms, and 11 - 13ms for release time c.f. Chapter 5.2.5. For prediction model that predicts four parameters jointly, the prediction errors are around 10ms for attack time and 160ms for release time. 10ms is smaller than all the audibility thresholds of attack time, although some of the audibility threshold of release time is

smaller than 160ms. Therefore, further development of the prediction model may need to focus more on release time. Overall, the audibility thresholds are at a similar level with the prediction error of most systems presented earlier, which means the prediction model has a good performance.

## 7.3 Conclusion

There are two experiments presented in this chapter. The investigation starts with the design of an optimisation algorithm. The two stage algorithm is designed to optimise the DRC parameters based on the perceptual aspects of the audio most affected by the DRC. Since the audio perceptual aspects are measured by the audio similarity model designed in Chapter 6, naturally the similarity performance is better if evaluated using the same model. Therefore evaluation is also completed using audio features that are related to DRC, such as crest factor and integrated loudness. The results indicate the efficiency of the algorithm. The second experiment is a listening test that is designed to investigate the audibility threshold of the ballistic parameters of the DRC. The results shows that the prediction models for single parameter have an error rate much lower than the audibility threshold. This backs up the efficiency of the prediction model design. In conclusion, both experiments presented in this chapter provide strong support for the proposed intelligent DRC control system from a perceptual perspective.

# Chapter 8

# Conclusions

This Thesis describes the research process involved in the design and evaluation of an innovative audio effect control system. The chapter summarises the main contributions by the author first. This chapter also aims to draw fundamental conclusions from the system design and experiments described throughout the Thesis. Suggestions for possible future development of this research is also included at the end.

## 8.1 Summary of contributions

The idea of using a reference audio to control an audio effect is first introduced in Chapter 3. It is an innovative approach in the research area of audio effect control. This modality, i.e. sound examples, received little attention in the significant body of work on intelligent audio production. The proposed method has three components, a feature extractor, a regression model for prediction, and a similarity model for evaluation and optimisation. The development of the feature extractor involves the analysis of the relations between low level audio features and the DRC. One discovery through the analysis is that each DRC parameter is best characterised using different feature sets. This finding is reasonable because each parameter corresponds to different aspects of the signal processing, however they affect the signal jointly as well. Therefore different feature sets are selected to predict different parameters, while there are also overlaps across the feature sets. For example, frequency domain statistical features, e.g. spectral centroid, are highly related to dynamics, which also shows high correlation with threshold and ratio of the DRC. Statistical features based on spectrogram are not highly correlated to attack time and release time, but the

statistical features based on MFCC are related to the ballistic parameters more strongly. This may be explained by the timbre information captured by MFCC, since attack time and release time also have their links to timbre.

Another contribution of the research is the design of handcrafted features. For isolated notes, the author has zoomed into the audio signal, and identified the ascending and descending parts of the audio that can be linked to attack time and release time. An energy related feature has been designed for ratio as well. More effort has been placed on the design of these parameters, because conventional features are proved to be able to predict threshold fairly accurately. Parameter estimation turned out to be harder for ratio, attack time and release time. Audio decomposition methods are applied to extend the handcrafted features to work with more complex audio materials, mono-instrument loops. When we process polyphonic music and predict parameter jointly, a deep neural network feature learning approach is proposed. The deep neural network has shown its advantage in learning the highly non linear functionality of the DRC as it does in other domains as well. The features designed or learnt in this research can be applied to similar problems in the future, for example, other non linear audio effects or audio synthesis parameter estimation given a target sound.

The Thesis contributes to the domain of audio similarity as well. There is little discussion on audio similarity differences induced by audio effects in the literature. This topic remains challenging and therefore definitely worth more attention. Not only can it be beneficial in the research area of Intelligent Music Production, more accurate similarity models can also be applied to other domains such as music information retrieval or auditory system analysis. A model focussing on the perceptual aspects of sound that can be altered by the DRC has been proposed in this research. The author has thoroughly evaluated the performance of each components of the model. Further research can be carried out to design a more generic model focussing on other perceptual aspects that were not considered in this Thesis.

A multi-stage random search based optimisation algorithm has been proposed in this Thesis as well. This research used the prediction model as a first step to perform global search, while the second stage is a local search based on the proposed similarity model. The proposed algorithm is a relatively new way to optimise the prediction model by introducing a second search stage.

A subjective test is included in this Thesis too. This experiment is designed to detect

the audibility threshold of the ballistics parameters of the DRC. The results support the prediction model by showing that most of the prediction errors are lower than the audible threshold. This listening test can be considered an initial one in a sequence of further studies. Thorough research can be done for the other parameters and in different compression modes.

Overall, the research has contributed to the discussion in the research area of Intelligent Music Production, as well as Audio Signal Processing. Many future research directions can be derived from this Thesis, as detailed in the next section.

## 8.2 Directions for Future Research

The author proposed an intelligent system targeting the DRC. This effect was chosen as it is a widely used audio effect and it is difficult to analyse because it has several stages of non linearity. The author hopes this Thesis would encourage similar research on other effects. It is also possible to jointly train a model to predict the parameters of an effect chain. This approach would have a more profound perceptual effect. There are some recent works in the area of Intelligent Audio Effect domain using end-to-end DNN to generate processed audio directly, instead of using it for feature learning. This is a different but very interesting approach. The author hopes this Thesis will provide useful analysis for this and other similar approaches.

The author has proposed handcrafted features for individual DRC parameters. This Thesis has moved to a more general feature learning approach in the later research stage. However, there are situations when handcrafted features provide more benefits than learnt feature embeddings. For example, effect control systems designed to work with simple audio materials. Handcrafted features provide the best performance for predicting single parameter on mono-instrument loops in Chapter 5. Additionally, using handcrafted features does not require training at least for the feature extraction component of the system. Therefore, designing handcrafted features is still interesting and valuable research. It is possible to dive into the signal level and find relative patterns. More efficient features can be designed and used in future research.

In terms of the feature learning model, the author has tried a more complex structure using three branches and a triplet loss to attempt feature learning [Sheng and Fazekas, 2018c]. In theory, this model should be more sensitive to the subtle difference than the twin

structured siamese model. However this approach did not provide significant improvement. Further research can be done on a more advanced model structure for feature learning targeting audio effects.

The current optimisation methods used the proposed audio similarity model as a measure of the random search method. Another possible approach is to use the audio similarity model as the optimisation function to train the feature learning model. Since the audio similarity model is not differentiable, potentially reinforcement learning can be applied. In this way, the features can become optimal for the perceptual aspects of the DRC or the targeted effect directly. The audio similarity model can be improved as well. As it is explained before, a model focussing on general audio perception can be a great benefit for future research.

The author hopes the listening test proposed in this Thesis could become the starting point of many similar works. It would be an interesting topic to find out what is the audible threshold for each audio effect and their parameters. These data can be used as important guidance for future research in Intelligent Music Production. These tests may provide deeper insights of the human auditory system and its computational simulation models.

Overall, there are many interesting works that can be completed in the domain of Intelligent Music Production. It is a relatively small community, but hopefully by writing this Thesis, the author encouraged other researchers to grow their interest and contribute to this research topic.

# Bibliography


Algorithms to measure audio programme loudness and true-peak audio level. `https://www.itu.int/rec/R-REC-BS.1770-4-201510-I/en`. Accessed Oct. 2015.

Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, Throsten Kastner, and Markus Cremer. Content-based identification of audio material using mpeg-7 low level description. In *ISMIR*, 2001.

Xavier Amatriain, Jordi Bonada, lex Loscos, Josep Lluís Arcos, and Vincent Verfaille. Content-based transformations. *Journal of New Music Research*, 32(1):95–114, 2003.

American_Standards_Association and Acoustical_Society_of_America. *American standard acoustical terminology:(including mechanical shock and vibration)*. American Standards Association, 1960.

Diego Ardila, Cinjon Resnick, Adam Roberts, and Douglas Eck. Audio deepdream: Optimizing raw audio with convolutional networks. In *Proceedings of the International Society for Music Information Retrieval Conference, New York, USA*, 2016.

Vaibhav Arora and Ravi Kumar. Probability distribution estimation of music signals in time and frequency domains. In *Proceeding of the 19th International Conference on Digital Signal Processing (DSP)*, pages 409–414. IEEE, 2014.

Jean-Julien Aucouturier and Francois Pachet. Music similarity measures: What's the use? 2002.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

Chris Baume, György Fazekas, Mathieu Barthet, David Marston, and Mark Sandler. Selection of audio features for music emotion recognition using production music. In *Audio*

*Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Juan Pablo Bello, Chris Duxbury, Mike Davies, and Mark Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556, 2004.

Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on speech and audio processing*, 13(5):1035–1047, 2005.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

Emmanouil Benetos, Sebastian Ewert, and Tillman Weyde. Automatic transcription of pitched and unpitched sounds from polyphonic music. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3107–3111. IEEE, 2014.

Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

Nancy Bertin, Roland Badeau, and Gaël Richard. Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–65. IEEE, 2007.

Nancy Bertin, Roland Badeau, and Emmanuel Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3): 538–549, 2010.

James C Bezdek and Nikhil R Pal. Some new indices of cluster validity. 1998.

Jeff Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. 1998.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Christopher M Bishop and Tom M Mitchell. *Pattern Recognition and Machine Learning*. Springer, 2014.

Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. Medleydb: A multitrack dataset for annotation-intensive mir research. In *in Proc. the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Jeroen Breebaart and Martin F McKinney. Features for audio classification. In *Algorithms in Ambient Intelligence*, pages 113–129. Springer, 2004.

Leo Breiman. Random forests. *Journal of Machine Learning Research*, 45(1):5–32, October 2001a. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL `http://dx.doi.org/10.1023/A:1010933404324`.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001b.

Gary Bromham, Dave Moffat, Mathieu Barthet, and György Fazekas. The impact of compressor ballistics on the perceived style of music. In *Audio Engineering Society Convention 145*. Audio Engineering Society, 2018.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

Prabir Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989.

Rui Cai, Lie Lu, Hong-Jiang Zhang, and Lian-Hong Cai. Highlight sound effects detection in audio stream. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–37. IEEE, 2003.

Emre Cakır and Tuomas Virtanen. End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input. *International Joint Conference on Neural Networks (IJCNN)*, 2018.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

Mark Brozier Cartwright and Bryan Pardo. Social-eq: Crowdsourcing an equalization descriptor map. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.

Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

David Clark. High-resolution subjective testing using a double-blind comparator. *Journal of the Audio Engineering Society*, 30(5):330–338, 1982.

Peter Daniel and Reinhard Weber. Psychoacoustical roughness: Implementation of an optimized model. *Acta Acustica united with Acustica*, 83(1):113–123, 1997.

Roger B Dannenberg. An intelligent multi-track audio editor. In *Proceedings of international computer music conference (ICMC)*, volume 2, pages 89–94, 2007.

Laurent Daudet and Bruno Torrésani. Hybrid representations for audiophonic signal encoding. *Signal Processing*, 82(11):1595–1617, 2002.

Brecht De Man, Kirk McNally, and Joshua D Reiss. Perceptual evaluation and analysis of reverberation in multitrack music production. *Journal of the Audio Engineering Society*, 65(1/2):108–116, 2017.

Sander Dieleman and Benjamin Schrauwen. Multiscale approaches to music audio feature learning. In *14th International Society for Music Information Retrieval Conference (ISMIR-2013)*, pages 116–121. Pontifícia Universidade Católica do Paraná, 2013.

Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6964–6968. IEEE, 2014.

Shyamala Doraisamy, Shahram Golzari, Noris Mohd, Md Nasir Sulaiman, and Nur Izura Udzir. A study on feature selection and classification techniques for automatic genre classification of traditional malay music. In *ISMIR*, pages 331–336, 2008.

Gianpaolo Evangelista. Pitch-synchronous wavelet representations of speech and music signals. *IEEE transactions on signal processing*, 41(12):3313–3330, 1993.

Harvey Fletcher. Auditory patterns. *Reviews of modern physics*, 12(1):47, 1940.

Jon Ford, Mark Cartwright, and Bryan Pardo. Mixviz: A tool to visualize masking in audio mixes. In *Audio Engineering Society Convention 139*. Audio Engineering Society, 2015.

Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2011.

Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *Proceedings of the International Symposium on Information Theory (ISIT)*, page 31. IEEE, 2004.

Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441, 1946.

Leonardo Gabrielli, Stefano Tomassetti, Stefano Squartini, Carlo Zinato, and Stefano Guaiana. A multi-stage algorithm for acoustic physical model parameters estimation. *arXiv preprint arXiv:1809.05483*, 2018.

Todor Ganchev, Nikos Fakotakis, and George Kokkinakis. Comparative evaluation of various mfcc implementations on the speaker verification task. In *Proceedings of the SPECOM.*

Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss. Digital dynamic range compressor designâĂŤa tutorial and analysis. *Journal of the Audio Engineering Society*, 60(6):399–408, 2012.

Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss. Parameter automation in a dynamic range compressor. *Journal of the Audio Engineering Society*, 61(10):716–726, 2013.

Brian R Glasberg and Brian CJ Moore. Prediction of absolute thresholds and equal-loudness contours using a modified loudness model. *The Journal of the Acoustical Society of America*, 120(2):585–588, 2006.

Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Music genre database and musical instrument sound database. *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, pages 229–230, 2003.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Toni Heittola, Anssi Klapuri, and Tuomas Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *Proceeding of the 10th International Society for Music Information Retrieval (ISMIR)*, 2009.

Perfecto Herrera, Xavier Amatriain, Eloi Batlle, and Xavier Serra. Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *International symposium on music information retrieval ISMIR*, volume 290, 2000.

John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

Marcel Hilsamer and Stephan Herzog. A statistical approach to automated offline dynamic processing in the audio mastering process. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

D.M. Huber and R.E. Runstein. *Modern Recording Techniques*. Audio Engineering Society Presents Series. Focal Press/Elsevier, 2010. ISBN 9780240810690. URL `https://books.google.co.uk/books?id=W9U7A-rSXtEC`.

Toshio Irino and Roy D Patterson. A time-domain, level-dependent auditory filter: The gammachirp. *The Journal of the Acoustical Society of America*, 101(1):412–419, 1997.

Roey Izhaki. *Mixing audio: concepts, practices and tools*. Taylor & Francis, 2013.

J. H. Jensen, M. G. Christensen, D. P. W. Ellis, and S. H. Jensen. Quantitative analysis of a common audio similarity measure. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):693–703, May 2009. doi: 10.1109/TASL.2008.2012314.

Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2):89–119, 2012.

Gregory Koch. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop. Vol. 2*, 2015.

Philipp Kolhoff, Jacqueline Preub, and Jorn Loviscach. Music icons: procedural glyphs for audio files. In *2006 19th Brazilian Symposium on Computer Graphics and Image Processing*, pages 289–296. IEEE, 2006.

Sunder Ram Krishnan, Mathew Magimai Doss, and Chandra Sekhar Seelamantula. A savitzky-golay filtering perspective of dynamic feature computation. *IEEE Signal Processing Letters*, 20(3):281–284, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789*, 2017.

Tom LH Li, Antoni B Chan, and A Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*. sn, 2010.

Wentian Li. Mutual information functions versus correlation functions. *Journal of statistical physics*, 60(5-6):823–837, 1990.

Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502, 2005.

Yuxiang Liu, Roger B Dannenberg, and Lianhong Cai. The intelligent music editor: towards an automated platform for music analysis and editing. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 123–131. Springer, 2010.

Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *ICME*, 2001.

Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *In International Symposium on Music Information Retrieval*, 2000.

Jörn Loviscach. Graphical control of a parametric equalizer. In *Audio Engineering Society Convention 124*. Audio Engineering Society, 2008.

Richard F. Lyon, Andreas G. Katsiamis, and Emmanuel M. Drakakis. History and future of auditory filter models. In *International Symposium on Circuits and Systems*, pages 3809–3812. IEEE, 2010.

Zheng Ma, Brecht De Man, Pedro DL Pestana, Dawn AA Black, and Joshua D Reiss. Intelligent multitrack dynamic range compression. *Journal of the Audio Engineering Society*, 63(6):412–426, 2015.

Donald M MacKay. Psychophysics of perceived intensity: A theoretical basis for fechner's and stevens' laws. *Science*, 139(3560):1213–1216, 1963.

Jacob A Maddams, Saoirse Finn, and Joshua D Reiss. An autonomous method for multitrack dynamic range compression.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

Marco Martínez and Joshua Reiss. End-to-end equalization with convolutional neural networks. *Proceedings of the 21th International Conference on Digital Audio Effects (DAFx-18)*, 2018.

M. A. Martínez Ramírez and J. D. Reiss. Stem audio mixing as a content-based transformation of audio features. In *19th International Workshop on Multimedia Signal Processing (MMSP), IEEE*, 2017.

Andrew Mason, Nick Jillings, Zheng Ma, Joshua D Reiss, and Frank Melchior. Adaptive audio reproduction using personalized compression. In *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology–Cinema, Television and the Internet*. Audio Engineering Society, 2015.

Sean McGrath, Alan Chamberlain, and Steve Benford. Making music together: An exploration of amateur and pro-am grime music production. In *Proceedings of the Audio Mostly 2016*, pages 186–193. ACM, 2016.

Stylianos Ioannis Mimilakis, Konstantinos Drossos, Tuomas Virtanen, and Gerald Schuller. Deep neural networks for dynamic range compression in mastering applications. In *Audio Engineering Society Convention 140*. Audio Engineering Society, 2016.

Thomas Mitchell. Automated evolutionary synthesis matching. *Soft Computing*, 16(12): 2057–2070, 2012.

Thomas J Mitchell and David P Creasey. Evolutionary sound matching: A test methodology and comparative study. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 229–234. IEEE, 2007.

Austin Moore, Rupert Till, and Jonathan Wakefield. An investigation into the sonic signature of three classic dynamic range compressors. In *Audio Engineering Society Convention 140*. Audio Engineering Society, 2016.

BCJ Moore and BR Glasberg. A model of loudness perception applied to cochlear hearing loss. *Auditory Neuroscience*, 3(3):289–311, 1997.

Brian CJ Moore. Development and current status of the âĂIJcambridgeâĂİ loudness models. *Trends in hearing*, 18:2331216514550620, 2014.

Brian CJ Moore and Brian R Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *The journal of the acoustical society of America*, 74(3):750–753, 1983.

Brian CJ Moore and Brian R Glasberg. A revision of zwicker's loudness model. *Acta Acustica united with Acustica*, 82(2):335–345, 1996.

Brian CJ Moore and Brian R Glasberg. Modeling binaural loudness. *The Journal of the Acoustical Society of America*, 121(3):1604–1612, 2007.

Meinard Müller. Fundamentals of music processing: Audio, analysis, algorithms, applications. 2015.

Dimitri Palaz, Mathew Magimai Doss, and Ronan Collobert. Convolutional neural networks-based continuous speech recognition using raw speech signal. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4295–4299. IEEE, 2015.

Elias Pampalk, Perfecto Herrera, and Masataka Goto. Computational models of similarity for drum samples. *IEEE transactions on audio, speech, and language processing*, 16(2): 408–423, 2008.

RD Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice. An efficient auditory filterbank based on the gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, volume 2, 1987.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. 2004.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

J. Pons, O. Slizovskaia, R. Gong, E. GÃşmez, and X. Serra. Timbre analysis of music audio signals with convolutional neural networks. *25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece*, 2017a.

Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2744–2748. IEEE, 2017b.

Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2744–2748. IEEE, 2017c.

Jun Qi, Dong Wang, Yi Jiang, and Runsheng Liu. Auditory features based on gammatone filters for robust speech recognition. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 305–308. IEEE, 2013.

K Ramamohan Rao and Ping Yip. *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

Joshua D Reiss. Intelligent systems for mixing multichannel audio. In *17th International Conference on Digital Signal Processing (DSP)*, pages 1–6. IEEE, 2011.

Derek W Robinson and R So Dadson. A re-determination of the equal-loudness relations for pure tones. *British Journal of Applied Physics*, 7(5):166, 1956.

Markus Röhl and Stefan Uppenkamp. Neural coding of sound intensity and loudness in the human auditory system. *JARO: Journal of the Association for Research in Otolaryngology*, 13(3):369–379, 2012.

DM Ronan, David Moffat, Hatice Gunes, Joshua D Reiss, et al. Automatic subgrouping of multitrack audio. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, 2015.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

Ryan Sarver and Anssi Klapuri. Application of nonnegative matrix factorization to signal-adaptive audio effects. In *Proc. DAFx*, pages 249–252, 2011.

Jan Schlüter and Sebastian Böck. Musical onset detection with convolutional neural networks. In *6th International Workshop on Machine Learning and Music (MML), Prague, Czech Republic*, 2013.

Jan Schluter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *Acoustics, speech and signal processing (ICASSP), 2014 ieee international conference on*, pages 6979–6983. IEEE, 2014.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

AT Schneider and JV Hanson. An adaptive dynamic range controller for digital audio. In *[1991] IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference Proceedings*, pages 339–342. IEEE, 1991.

Emery Schubert, Joe Wolfe, and Alex Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *International Conference on Music Perception and Cognition, North Western University, Illinois*, pages pp. 112–116, 2004.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978.

Yang Shao, Zhaozhang Jin, DeLiang Wang, and Soundararajan Srinivasan. An auditory-based feature for robust speech recognition. In *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4625–4628, 2009.

Di Sheng and György Fazekas. Automatic control of the dynamic range compressor using a regression model and a reference sound. In *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

Di Sheng and György Fazekas. Feature design using audio decomposition for intelligent control of the dynamic range compressor. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 621–625. IEEE, 2018a.

Di Sheng and György Fazekas. Feature selection for dynamic range compressor parameter estimation. In *Audio Engineering Society Convention 144*. Audio Engineering Society, 2018b.

Di Sheng and György Fazekas. Using triplet network for the intelligent control of audio effects. In *Proceedings of Digital Music Research Network Workshop 2018, (DMRN+13)*, 2018c.

Kai Siedenburg and Simon Doclo. Iterative structured shrinkage algorithms applied to stationary/transient separation. In *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

Kai Siedenburg and Monika Dörfler. Persistent time-frequency shrinkage for audio denoising. *Journal of the Audio Engineering Society*, 61(1/2):29–38, 2013.

Malcolm Slaney. An efficient implementation of the patterson-holdsworth auditory filter bank. *Apple Computer, Perception Group, Tech. Rep*, 35(8), 1993.

Ryan Stables, Sean Enderby, BD Man, György Fazekas, Joshua D Reiss, et al. Safe: A system for the extraction and retrieval of semantic audio descriptors. 2014.

Charles J Stone, JH Friedman, L Breiman, and RA Olshen. Classification and regression trees. *Wadsworth International Group*, 8:452–456, 1984.

Bob L Sturm. A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pages 29–66. Springer, 2012a.

Bob L Sturm. Two systems for automatic music genre recognition: What are they really recognizing? In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 69–74. ACM, 2012b.

D. Tardieu, E. Deruty, C. Charbuillet, and G. Peeters. Production effect: Audio features for recording techniques description and decade prediction. In *in Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx- 11), Paris, France.*, 2011.

Michael J Terrell, György Fazekas, Andrew JR Simpson, Jordan Smith, and Simon Dixon. Listening level changes music similarity. *ISMIR*, 2012.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary detection in music structure analysis using convolutional neural networks. In *ISMIR*, pages 417–422, 2014.

M. Unoki, T. Irino, B. Glasberg, BCJ. Moore, and RD. Patterson. Comparison of the roex and gammachirp filters as representations of the auditory filter. *The Journal of the Acoustical Society of America*, 120(3):1474–1492, 2006.

Vincent Verfaille, Catherine Guastavino, and Caroline Traube. An interdisciplinary approach to audio effect classification. Citeseer, 2006a.

Vincent Verfaille, Udo Zolzer, and Daniel Arfib. Adaptive digital audio effects (a-dafx): A new class of sound transformations. *IEEE Transactions on audio, speech, and language processing*, 14(5):1817–1831, 2006b.

Vice. Meet the nz producers making hits from their bedrooms, 2018. URL `https://www.vice.com/en_au/article/9k8kxv/meet-the-nz-producers-making-hits-from-their-bedrooms`.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

Wil M Wagenaars, Adrianus J Houtsma, and Ruud A van Lieshout. Subjective evaluation of dynamic compression in music. *Journal of the Audio Engineering Society*, 34(1/2): 10–18, 1986.

Siying Wang. *Computational Methods for the Alignment and Score-Informed Transcription of Piano Music*. PhD thesis, Queen Mary University of London, 2017.

Siying Wang, Sebastian Ewert, Simon Dixon, Siying Wang, Sebastian Ewert, and Simon Dixon. Identifying missing and extra notes in piano recordings using score-informed dictionary learning. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(10):1877–1889, 2017.

Matt Welsh, N Borishov, Jason Hill, Rob von Behren, and Alec Woo. Querying large collections of music for similarity. Technical report, Technical report, University of California, Berkeley, CA, 1999.

Mark Wendl and Hyunkook Lee. The effect of dynamic range compression on loudness and quality perception in relation to crest factor. In *Audio Engineering Society Convention 136*, Apr 2014. URL `http://www.aes.org/e-lib/browse.cfm?elib=17168`.

Thomas Wilmering, György Fazekas, and Mark B Sandler. High-level semantic metadata for the control of multitrack adaptive digital audio effects. In *Audio Engineering Society Convention 133*. Audio Engineering Society, 2012.

Thomas Wilmering, György Fazekas, and Mark B Sandler. Audio effect classification based on auditory perceptual attributes. In *Audio Engineering Society Convention 135*. Audio Engineering Society, 2013.

Hao-Chun Yang, Fu-Sheng Tsai, Yi-Ming Weng, Chip-Jin Ng, and Chi-Chun Lee. A triplet-loss embedded deep regressor network for estimating blood pressure changes using prosodic features. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6019–6023. IEEE, 2018.

Wei Yang and Igor Zurbenko. Kolmogorov–zurbenko filters. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3):340–351, 2010.

Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Udo Zölzer, Xavier Amatriain, and Daniel Arfib. *DAFX: digital audio effects*, volume 1. Wiley Online Library, 2002.