A Study in Violinist Identification using Short-term Note Features

Yudong Zhao

A thesis 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
United Kingdom

2022

Statement of Originality

I, Yudong Zhao, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

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:

Abstract

The perception of music expression and emotion are greatly influenced by performer's individual interpretation, thus modelling performer's style is important to music understanding, style transfer, music education and characteristic music generation. This Thesis proposes approaches for modelling and identifying musical instrumentalists, using violinist identification as a case study.

In violin performance, vibrato and timbre play important roles in players' emotional expression, and they are key factors of playing style while execution shows great diversity. To validate that these two factors are effective to model violinists, we design and extract note-level vibrato features and timbre features from isolated concerto music notes, then present a violinist identification method based on the similarity of feature distributions, using single feature as well as fused features. The result shows that vibrato features are helpful for the violinist identification, and some timbre features perform better than vibrato features. In addition, the accuracy obtained from fused features is higher than using any single feature.

However, apart from performer, the timbre is also determined by musical instruments, recording conditions and other factors. Furthermore, the common scenario for violinist identification is based on short music clips rather than isolated notes. To solve these two problems, we further examine the method using note-level timbre features to recognize violinists from segmented solo music clips, then use it to identify master players from concerto fragments. The results show that the designed features and method work very well for both types of music. Another experiment is conducted to examine the influence of instrument on the features. Results suggest that the

selected timbre features can model performers' individual playing reasonably and objectively, regardless of the instrument they play.

Expressive timing is another key factor to reflect individual play styles. This Thesis develops a novel onset time deviation feature, which is used to model and identify master violinists on concerto fragments data. Results show that it performs better than timbre features on the dataset.

To generalise the violinist identification method and further improve the result, deep learning methods are proposed and investigated. We present a transfer learning approach for violinist identification from pre-trained music auto-tagging neural networks and singer identification models. We then transfer pre-trained weights and fine-tune the models using violin datasets and finally obtain violinist identification results. We compare our system with state-of-the-art works, which shows that our model outperforms them using our two datasets.

Acknowledgements

First and foremost, I would like to thank my primary supervisor, Prof. Mark Sandler, for opening me the door to "music information retrieval". Although I was a Master student majoring in electronic engineering and FPGA programming, he gave me the opportunity of being part of the Centre for Digital Music (C4DM) and meeting people in this fantastic group. Over the past four years, with his invaluable guidance, continuous encouragement and support, I have strengthened myself personally and professionally. He always supported my research with his sage advice and patient instruction. I have significantly benefited from his scientific rigour, broad knowledge and vast research experience.

I want to express my special appreciation to Dr. György Fazekas for his impressive competence in all technical matters and constant kindness and patience in answering all the trivial questions I asked. He was always there to advise me when I struggled with an experiment and didn't know how to solve it or when I finished a paper draft and didn't know how to revise it. I would also like to thank Prof. Josh Reiss for his help with music expression modelling and guidance during my tough starting days.

I would like to express my great gratitude to all the people who supported my research in various ways. Thanks to Dr. Mi Tian for her guidelines on Ph.D application and Vamp plugin development; Dr. Beici Liang for sharing her experience on research presentation and programming skills; Dr. Ken O'Hanlon for the helpful discussion on music signal processing and harmonic

visualiser; Dr. Keunwoo Choi for the tutorials of machine learning and deep learning on Music Information Retrieval; Dr. Chris Cannam for the instruction of data annotation using sonic visualiser; Dr. Claudia Friz for sharing me the solo violin recordings and the experience on violin classification; Dr. Simin Yang for the advice on Ph.D thesis writing; Dr. Emmanuouil Benetos for the discussion on data pre-processing and data split strategy; Dr. Changhong Wang for her very careful attitude on the collaborated research work; Dr. Shengchen Li for sharing his research experience on music expressive timing; Dr. Luwei Yang for sharing his work on vibrato detection; Dr. Di Sheng for introducing her experiences on audio feature development. Special thanks to the anonymous reviewers of the papers I have submitted for the helpful feedback.

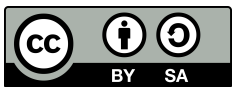
I am grateful for all financial support for my research, publications and conference presentations from various funding parties, including China Scholarship Council (NO. 201706020141), EPSRC C4DM Travel Funding and EPSRC Fusing Semantic and Audio Technologies for Intelligent Music Production and Consumption (EP/L019981/1).

Big thanks to everyone in C4DM: An, Alessia, Andrew, Charis, Dalia, Dan, Daniel, Dave, Delia, Fred, Giulio, Huy, Jeff, Jianing, Jiawen, Lele, Louise, Matthias, Matthieu, Mike, Nick, Qiuqiang, Rishi, Saurjya, Sebastian, Simon, Syde, Tim, Vinod, Yukun and so on. Particular thanks go to my flatmates and friends Ye Sun and Man Zhang, who made me never feel lonely in a foreign country, especially during the difficult period of the Covid-19 pandemic. I will never forget the hotpots and Chinese Cuisine we made on Chinese New Year's Eve and other traditional holidays.

A special thanks go to my girlfriend, Jinyu Zhan. There is so much I want to say, but so little can be expressed. Although we have not been in the same country in the last few years, I always feel she is close to me. Whenever I feel depressed or unmotivated, she is always the first to comfort me with a

lot of patience, constant help and encouragement. Last but not least, with great emotion, I would like to thank my parents, Qiuju Wang and Wei Zhao, who have always given me so much love and kindness, always supported and encouraged me, and I am so proud to be their son. All of this would not have been possible without their support and encouragement.

Licence



Contents

List of Figures

List of Tables

List of abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ACC	All Concerto Clips
ASC	All Solo Clips
CNN	Convolutional neural network
CQT	Constant-Q Transform
CRNN	Convolutional recurrent neural network
CV	Cross validation
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DNN	Deep neural network
f0	Fundamental frequency
FFT	Fast Fourier transform
FN	False negative
FP	False positive
GMM	Gaussian Mixture Model
IVN	Isolated Vibrato Notes
IPID	Instrumental performer identification
JS	Jensen-Shannon
KDE	Kernel density estimation
KNN	K-nearest neighbours

KL	Kullback-Leibler
LSTM	Long Short-Term Memory
LOGOCV	Leave one group out cross-validation
MIR	Music information retieval
MFCCs	Mel-frequency cepstral coefficients
OTD	Onset Deviation
PDF	Probability density function
ReLU	Rectified Linear Unit
RMS	Root mean square energy
SB	Spectral Bandwidth
SC	Spectral Centroid
SCT	Spectral Contrast
SCC	Selected Concerto Clips
SID	Singer identification
SSC	Selected Solo Clips
SVM	Support Vector Machine
TP	True negtive
TP	True positive
VID	Violinist identification
Zero-crossing Rate	ZCR

Chapter 1

Introduction

1.1 Motivation

As explained in Britannica [1], music is defined as "art, concerned with combining vocal or instrumental sounds for beauty of form or emotional expression, usually according to cultural standards of rhythm, melody, and, in most Western music, harmony". With the increasing diversity of music types and styles, people's music demands and preferences have become more individualistic. For example, when people want to nap, a calming piece of music may help them fall asleep quickly; when they feel down or stressed, listening to rock music can make them feel refreshed again. Therefore, music expression is essential in conveying emotion and resonating with listeners.

It is well-known that music expression is generally related to two interdependent factors: the structure established by the composer and the interpretation presented by the performer [1]. Generally, pitch and rhythm created by music composers or producers are considered typical music structural characteristics, which determine the melody and tempo of a piece of music. In classical music, composers rarely change the music structure after

[1] https://www.britannica.com/art/music (Accessed 20 August 2022)

they publish the music. But, a particular piece's expression and emotion can vary due to differences in tempo, sound intensity and playing technique. For example, although beat has been referred to above as a parameter of musical structure, it can be sped up or slowed down by the performer in a very flexible manner. Apart from this, other individual cues also play essential roles in music expressiveness, such as timbre, articulation, vibrato, tone attacks and tone decays [2]. Adjusting these factors allows a given musical piece to be performed in completely different styles; a lovely music piece can be played dully, while a rousing piece can be played softly. From the author's point of view, the expressive factors are thus more decisive and influential in the delivery of musical expression, where subtle emotional variations are often introduced by the performers rather than composers.

It is challenging to characterise musical expressions from audio signals due to the interplay of structural and interpretative factors. Still, the music expression analysis is valuable in applications like automatic music transcription, music playlist recommendation, computer-aided music education, and expressive music generation. Since the expressive diversity of a given musical piece mainly depends on parameters such as timbre, loudness or articulation [3], which are usually determined by the interpretation of the performer, an in-depth analysis of the performer's style is essential to the musical expression study. Different performers will perform differently on the same music piece since each has preferences, personality, educational background, physical conditions, etc. The interpretations by performers lead to many different versions of the same musical composition, giving the listener more musical enjoyment.

Most listeners have their favoured instrumentalists or singers, meaning that these performers have a unique and appealing style that makes their performances popular and artistically valuable. Unfortunately, some virtuoso performers have passed away, or their playing level has declined with age or

health problems. Listening to CD recordings or watching videos is the only way to enjoy their performances, but it is a shame that if there is a piece that a deceased performer never played during the lifetime, we will not be able to hear that performer play it. For example, Heifetz never played the famous Chinese piece "Liang Zhu" during his lifetime, and we will never have the opportunity to hear it played in Heifetz's style. Therefore, to make up for this regret to some extent, it is crucial to understand and describe the style of virtuoso performers using objective music descriptors and then verify the effectiveness of such descriptors by modelling and identifying performers. After that, we can morph known playing styles into new music compositions or transfer interpretation from one existing to another. In addition, if we can obtain valid musical indicators to model the performers' styles, it will be helpful to reproduce the characteristics of different virtuoso performers on any musical piece. Moreover, we can also make it possible to permutate and combine various performances played by different performers into a single music clip, which will be very useful for music editing and production.

In musical performances, singers create music with their singing voices, while instrumentalists produce music with their instruments. Thus the latter's performances are not so closely tied to performers as the former's. Moreover, instrumentalists' style modelling and identification are more challenging, which has received less attention than singer identification due to the lack of large-scale public datasets. Among all instruments, the violin undoubtedly plays a vital role in both classical and modern music, and many great violinists deserve to be studied in depth. If we could quantitatively describe and model the characteristic style of the performer, it would help music producers to produce violin pieces with a wide range of styles. We can also measure the similarity of playing styles between a violin student and a famous violinist, which can help improve the student's playing skills.

Based on the above problems and conjectures, this Thesis proposes violin-

ist modelling and identification approaches that apply to different scenarios. We first construct two types of datasets containing solo music and concerto music, respectively, to validate our proposed methods. Then, audio features are designed and extracted based on the domain knowledge of violin playing, which is used to describe the playing styles of violinists. Next, several statistical models are used to model the style of violinists, and the similarity of such models is calculated to identify violinists. In addition, considering the strong performance of deep learning in other related fields, we also apply deep learning methods to classify violinists to compensate for the lack of generalisation and less-than-optimal results of the above mechanisms.

The next section will review the musical performer identification methods in previous studies. The outline of this Thesis and associated publications will be introduced in Section 1.3 and Section 1.4, respectively.

1.2 A Review of Performer Identification

Since a performer's style greatly influences the expression of a given music piece, style modelling and performer identification have been studied for a long time. In the early days, the primary method of analysing performers' styles was statistical and mathematical modelling of audio features. Repp [4] characterised temporal commonalities and differences among famous pianists' interpretations of a well-known piece using a variety of statistical analyses and demonstrated the individuality of two legendary pianists. Stamatatos [5] presented a comparison of features for discriminating 22 pianists playing the same piece based on a series of statistical experiments. The result suggested that the "average performance" effectively recognises individual performances, while "extreme" performances have the lowest discrimination. Similarly, Bresin [6] analysed articulation strategies from five pianists' performances by measuring the mean and variance of inter-onset-interval (IOI)

times.

In recent years, machine learning has been widely applied for performer identification. Stamatatos and Widmer [7] proposed a set of features like time deviation and melody lead [8] that capture aspects of pianists' style, then illustrated how a machine could distinguish and classify music performers by their performance style. Saunders et al. [9] applied string kernels to the problem of recognising famous pianists by style. Performer playing characteristics were obtained from changes in beat-level tempo and loudness, which derived from a general performance alphabet and represented pianist's performances. Ramirez et al. [10] developed a machine learning approach to identify Jazz saxophonists by analysing individual notes' pitch, timing, amplitude and timbre. Apart from instrumentalist identification, some works are also based on singer recognition. Nadine Kroher [11] investigated a robust system of modelling the singer's typical performance style using vibrato, timbre and statistical performance descriptors. Audio feature learning methods are also frequently used for singer identification in many other researches [12, 13, 14].

Deep neural networks (DNN) greatly outperform hand-crafted feature engineering methods for singer identification (SID) in this deep learning era. Zhang [15] reported outstanding SID results with 0.99 f1-score using DNN on the Artist20 [16] dataset. Also, Nasrullah [17] proposed a singer classification method with convolutional recurrent neural networks. In addition, source separation is applied on SID and showed good results [15, 18]. However, there are not many instrumentalist identification systems implemented by deep learning methods, possibly due to the lack of a large-scale training dataset.

In addition, we propose methods based on subjective experiments to discuss the personal factors influencing performers' styles. For example, Gingras et.al [19] explored the influence of both listener and performer's level of expertise on performer identification, which suggested that the performer's

level of expertise had a more significant impact on the result. In addition, the performer's self-identification was also investigated in [20], and the results showed that temporal cues are essential for performers to identify themselves.

In particular, there are prior works on violin expression analysis and violinist identification. Li et al. [21] developed a dataset containing 11 expressive characteristics; features like duration, dynamics and vibrato features are extracted to classify expressions using Support Vector Machine (SVM). Ramirez et al. [22] built a Celtic violinist classifier using the machine learning method. They extracted pitch, timing and amplitude features representing both note-level characteristics and broader musical context. Molina et al. [23] proposed an approach to identify violinists on monophonic audio recordings using a musical trend-based model. Shih et al. [24] used articulation and energy features to compare the playing styles of Heifetz and Oistrakh, arguably the most talented violinists in the world.

However, most previous studies are based on the dataset of solo music rather than common violin performance scenarios (such as concertos, sonatas, and repertoire), which limited their application and generalisation. In addition, previous studies have focused more on classifying professional performers or a limited number of virtuosos and less on identifying more than five virtuosos. In addition, in earlier works, essential music descriptors such as vibrato and expressive timing have rarely been used for violinist identification. It isn't easy to know how effective these descriptors are for identifying violinists. Finally, although deep learning has been widely used in many fields and has shown powerful capabilities, there is little research on violinist identification based on deep learning methods. The approach presented in this Thesis will attempt to address these issues and provide corresponding conclusions based on our experiments.

1.3 Thesis Structure and Contributions

Chapter 2 Background

This Chapter reviews the technical background of this Thesis. It starts by introducing the concept of musical expression and the importance of the performer's interpretation of the musical expression. As a performer's style is often achieved by personal application of instrumental playing techniques, we introduce the fundamental skills of violin playing. Next, since the main aim of this research is to develop a violinist identification system, which involves procedures including audio feature extraction, classification model construction and result evaluation, the relevant background of each method is reviewed in detail in this Chapter.

Chapter 3 Dataset Construction

This Chapter focuses on the datasets we constructed for evaluating the proposed violinist identification algorithms. Based on the concertos recordings of the master players, we have created a dataset of isolated vibrato notes, a dataset of short selected music clips and a dataset of constant violin concerto clips, which will be applied in Chapter 4, 6 and 7 respectively. In addition, another dataset based on solo violin playing is constructed and presented in this Chapter, which not only provides additional scenarios for violinist identification but also allows us to verify the validity of the timbre features in Chapter 5.

Chapter 4 Violinist Identification Using Isolated Notes

This Chapter presents an original work on violinist identification using isolated musical notes. Two categories of audio features, including vibrato features and timbre features, are designed and extracted, and the global distribution of each feature is obtained to model the performer's style. After the violinist identification is achieved by using each feature,

a feature fusion method is proposed for merging the discrete knowledge, which is different kinds of audio features in our case, to yield a more comprehensive representation of the performer's characteristic playing. Finally, the effectiveness of different statistical models for identifying violinists is compared and discussed.

Chapter 5 The Effectiveness of Timbre Features for Identifying Violinists

This Chapter reveals the effectiveness of proposed timbre features for modelling and identifying violinists. Although the previous Chapter's results show that timbre features perform well for classifying violinists, musical timbre is intuitively influenced by other factors such as instruments or recording conditions. The timbre features are thus likely to model the instrument's characteristics or acoustic recording rather than the performer's style. To address this issue, we verify the effectiveness of timbre features in modelling players' characteristics by conducting violinist identification experiments and violin identification experiments. The results show that the designed features are beneficial in identifying violinists, regardless of which violin is played.

Chapter 6 Violinist Identification Using Short Music Clips

This Chapter proposes a violinist identification method based on short music clips. It starts with the development of expressive timing features, and their performances in recognising violinists are firstly evaluated. Then, the timbre features presented in the previous chapters are extracted from the short music clip dataset, and these features are applied to identify violinists. Finally, the two kinds of features are fused to identify the violinist, producing better discrimination results than using a single kind of feature.

Chapter 7 Transfer Learning for Violinist Identification

This Chapter aims to investigate a deep learning method for identi-

fying violinists based on a limited dataset, which is the first time, to our knowledge, that a deep learning method has been applied to the field. We propose a transfer learning approach that uses pre-trained models for other MIR tasks and then fine-tunes those models on our constructed concerto and solo datasets. Comparing the results of transfer learning and 'training from scratch' shows that transfer learning is very effective for this task. This Chapter also compares the performance of deep learning methods and those presented in the previous Chapters.

Chapter 8 Conclusion

This Chapter concludes this Thesis and indicates some potential directions for future work.

1.4 Associated Publications

This Thesis covers the work carried out by the author between September 2017 and March 2022 at Queen Mary University of London. The majority of the work presented in this Thesis has been published in peer-reviewed conferences.

- Yudong Zhao, György Fazekas, and Mark Sandler. "Identifying Violinist using note-level vibrato features," 2019 SEMPRE Autumn Conference, 2019.9.

- Yudong Zhao, György Fazekas, and Mark Sandler. "Identifying Master Violinists Using Note-level Audio Features", 17th Sound and Music Computing Conference, 2020.7.

- Yudong Zhao, Changhong Wang, György Fazekas, Mark Sandler. "Violinist identification based on vibrato features", European Signal Processing Conference (EUSIPCO), 2021.

- Yudong Zhao, György Fazekas, Mark Sandler, "Violinist identification based on note-level timbre feature distribution", International Conference on Acoustics, Speech, Signal Processing (ICASSP), 2022.
- Yudong Zhao, György Fazekas, Mark Sandler, "Transfer learning for violinist identification", European Signal Processing Conference (EUSIPCO), 2022

Chapter 2

Background

2.1 Introduction

This Chapter reviews the technical background of this Thesis. The first section is dedicated to introducing musical expression, followed by a detailed description of the factors influencing it. Then we briefly introduce the violin structure and basic playing techniques in Section 2.3. Next, a special focus is given in Section 2.4 on the audio features that have been used to describe music characteristics, and common feature extraction methods are reviewed. It is followed by an overview of statistical models and audio similarity algorithms in Section 2.5, which are separately applied to represent and analyse musical expressions. As this Thesis widely uses machine learning methods, we discuss their definition and application on Music Information Retrieval (MIR) in Section 2.6. Finally, the evaluation metrics are presented in Section 2.7.

2.2 Expressive Music Performance

One of the key reasons why music captivates listeners is that it can express emotion and resonate with them [25]. How music evokes listeners' feelings have been studied from many perspectives, and it provides a rich field for studying music perception and music production [26]. However, it is challenging to assess the exact contribution of individual musical factors (e.g., timbre, timing, dynamic etc.) to emotional expression because many of them are intercorrelated. One method to address this problem is to manipulate the cues in music by synthesising variants of given music independently and systematically [27]. However, the music expression generally depends on three main factors: the structure established by the composer (e.g. mode, pitch, or dissonance), the arrangement and sound effect produced by producers, and the interpretation of the performer (e.g., speed, loudness) [1]. These three factors are introduced separately in this section.

2.2.1 Music Structure and Production

Musical structure is broadly defined by the parameters of a musical piece laid down in the score [26], and the most general structural characteristics of Western music are pitch and rhythm. Schenker's music theory [28] considers the melodic and harmonic organisation as a progressively more complex series of elaborations on a simple base, with musical expression arising gradually from note to note. Furthermore, music without precise time control is considered deficient because it lacks the property of rhythm [29]. Some studies provide evidence that ordinary listeners can accurately and immediately use changes of mode and tempo to perceive music expressions and emotions [1]. What is commonly believed is that fast tempo and major mode always correspond to happiness and cheeriness, whereas slow tempo and minor mode frequently link to sadness or sorrow [30]. Dissonance is another essential

structural parameter that correlates with the activity of human paracortical and neocortical areas. It also reveals the neural basis of the human emotional response to music [31].

In addition to the musical structure defined by the composer, musical production greatly influences musical expression, especially in modern music. The same song can be presented in various styles if produced by different producers or mixed with different instruments. For example, both versions of *"Love Story"* are sung by *Taylor Swift*, but different mixes and arrangements present different sensations [1,2]. The music production phase, which includes "recording" and "editing", focuses primarily on capturing the desired sound quality and audio effects, which largely influence the musical expression. In today's music post-processing, digital audio effects play a crucial role in shaping the desirable sound by varying timbre, dynamic range and instrumentation to present different musical expressions and emotions [32].

2.2.2 Individual Styles of Performer

Although musical structure and production can influence music expression, the expressive richness of a given musical piece strongly depends on the interpretation of the performer [1]. Since most audiences learn about music by listening to recordings or going to concerts without reading the scores beforehand, the performer's interpretations dominate people's music perception to a great extent. In addition, the performer introduces many micro-variations of musical parameters (e.g., tempo, loudness, or articulation) to enhance the emotional expression of the original musical score [33]. For example, although pitch is a parameter of musical structure, the pitch of a long sustain can be slightly altered (e.g., vibrato) by a performer to enhance the dynamics of the music. The manipulation of these musical parameters appears to be

[1]https://www.youtube.com/watch?v=mNLVMDF9mUo (Accessed 20 August 2022)
[2]https://www.youtube.com/watch?v=etywYG4ZSvg (Accessed 20 August 2022)

particularly important for musical expression.

Furthermore, professional performers, especially master instrumentalists, prefer to interpret a given music excerpt in their styles. For instance, Jung [34] analysed the playing styles of three master violinists: Jascha Heifetz, David Oistrakh and Joseph Szigeti, and mentioned that the precision and fast tempo in Heifetz's performance give listeners the feeling of "cold" and "unemotional." In contrast, Oistrakh was described as "warm" and "capable of communicating emotional feelings." Other important cues influencing musical expression include sound level, pitch, articulation, timbre, vibrato, tonal attack and decay [2]. These cues convey different kinds of expression intentions and are decided mainly by how performers apply their playing styles.

In summary, although expressive musical performances are affected by musical composition and production, the subtlety of musical expression is generally given by the performer's interpretation. The ambiguity in musical notation gives the performer considerable freedom in deciding how to interpret the content, allowing a specific music piece to be performed in various styles by different performers. The interpretation also refers to performers' individualistic preference of a clip according to their ideas and musical intentions [35]. In violin performance, the applications of playing techniques can produce various expressive styles. For example, vibrato is produced by the movement of fingers on the fingerboard; different bowing gestures create different timbres. Therefore in the next section, we will briefly introduce the structure and basic violin playing techniques.

2.3 Brief Introduction of Violin Playing

Although discussing how to play the violin is beyond this Thesis's scope, the application of playing techniques heavily influences the characteristics of violin playing. In this section, we first introduce the structural components

of the violin and outline basic violin playing techniques.

2.3.1 Violin Structure

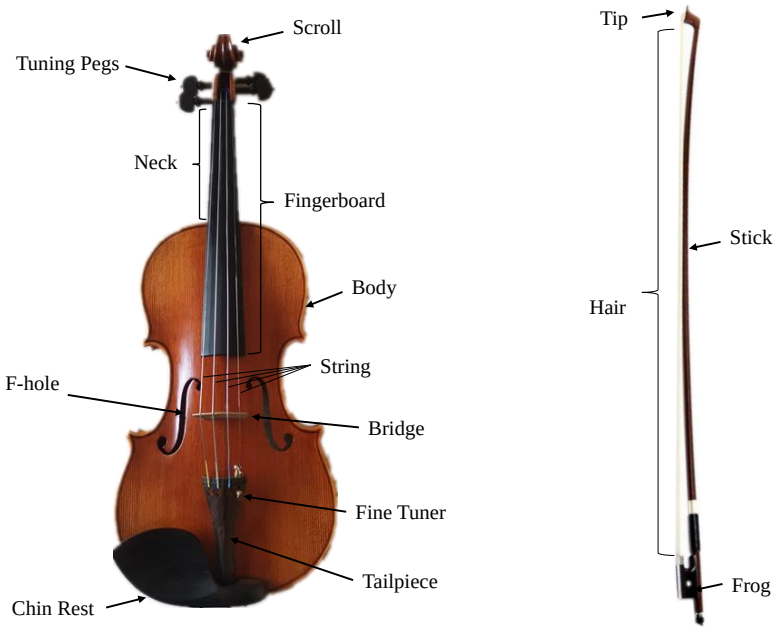


Figure 2.1: The parts of a violin(left) and a bow(right).

Violin is a wooden string instrument commonly played by drawing a bow across strings. Figure 2.1 depicted a standard modern violin and a bow from the front view with labelled parts. The scroll is an important decoration placed at the top of the violin and farthest from the performer. Immediately below the scroll is the tuning box, with four tuning pegs housed in it, which adjust the strings' pitch. The black wood attached to the neck is the fingerboard, and the player can press the strings at different positions on the fingerboard to change the length of vibration of the strings and thus produce different pitches. The other end of the strings is anchored on the tailpiece through a bridge. The function of the bridge is to transmit the strings' vibration to the violin body and then spread the sound outside through the

(a) (b)

Figure 2.2: Samples of press string by left hand (a) and bow hold by right hand (b).

F-hole. There is also a fine tuner on the tailpiece, which is used to tune strings subtly. Beginners typically use fine tuners on all strings, whereas advanced players only use fine tuners on the E-string [36]. The violin has four strings, usually tuned in perfect fifths with notes G3, D4, A4, and E5. The Chin Rest is clamped to the violin body and helps the player's chin to hold the violin properly.

The violin bow consists of a flexible wooden stick, and a bundle of horsehair fixed to each end of the stick. The performer's near-end is named "frog", and the far end is called the "tip". The player holds the bow at the frog and controls the bowing gestures such as position, speed, and tilt on the string to produce a different sound.

2.3.2 Basic Playing Technique

The violin is usually performed using both hands together (except for left-hand pizzicato). The thumb of the left hand is placed on one side of the fingerboard; the other four fingers can press the strings in different positions. The pitch can be varied by changing the length of the string vibration. Placing the fingers closer to the player's body will produce a higher pitch, while pressing the string further away from the player will create a lower pitch.

Players can also shift the entire left hand towards the body to reach much higher pitches, called *shifting*. The common gesture of the left hand in a neutral position is shown in Figure 2.2(a). The right hand is responsible for drawing the bow over the strings, making the strings vibrate to produce sound. The basic gesture of holding a bow is shown in Figure 2.2(b), but it can be varied when playing different musical pieces.

In the following subsections, we will introduce a typical left-hand playing technique (vibrato) and basic bowing techniques , which will help to understand the designing of hand-crafted audio features that are used to model the violinist's playing style.

2.3.2.1 Vibrato

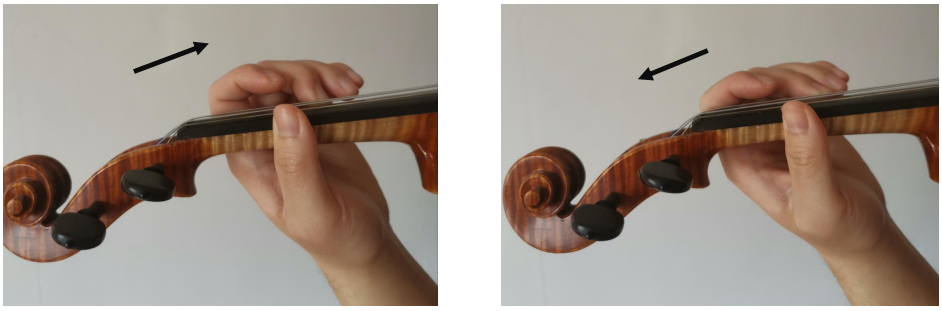


Figure 2.3: Finger movement when playing the vibrato.

One of the important left-hand techniques in violin playing is vibrato. It is an expressive tool that enhances the mood of a note or phrase and creates a warmth or richness of music. The performer creates vibrato by slightly changing the length of the string, which is produced by rocking the finger

from the wrist or arm (as shown in Figure 2.3). Itzhak Perlman, a famous master violinist, classifies violin vibrato as arm vibrato, hand vibrato and finger vibrato based on the origin of the gesture; or category it as slow vibrato, narrow vibrato and wide vibrato based on the dynamics of the gesture [3]. [37] also illustrates the rates and widths are two critical elements of performing vibratos, which are realised by modifying the speed and extent of the finger moving. Performers can apply different vibrato styles to present their desired sound by modifying the speed and amplitude of the vibrato. This property will be used for the vibrato feature development, and further analysis will be presented in Section 2.4.2.2.

2.3.2.2 Bowing Technique

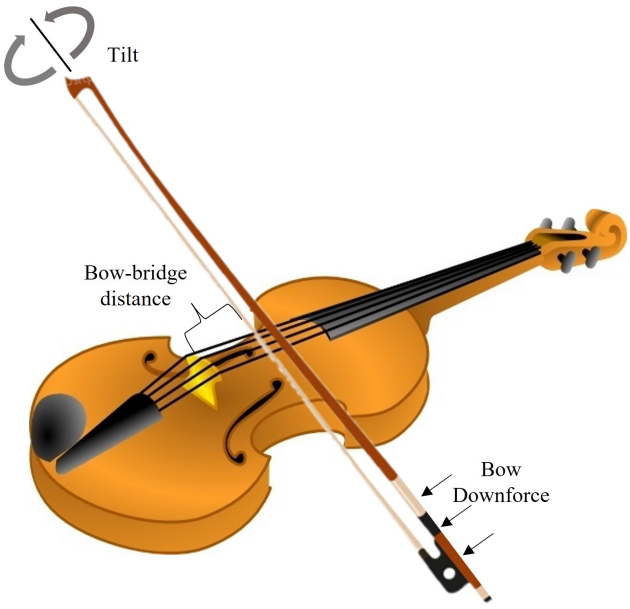


Figure 2.4: The diagram of bowing gestures.

When playing the violin, players can adjust bowing gestures like bow speed, bow downforce, tilt, and bow-bridge distance to produce their de-

[3]https://www.youtube.com/watch?v=SodZCoSBIR0 (Accessed 20 August 2022)

manded sound in different timbre, volume, or tone quality [36]. As shown in Figure 2.4, bow-bridge distance is the distance between the bridge and the contact point of the hair and string. Bow downforce is given by the right hand and arm, which can increase or decrease the amount of hair in contact with the string. Bow tilt is the axial rotation of the bow, which is adjusted to change the contact area. As for the bow speed, intuitively, it is the speed at which the bow moves across the string. In practice, performers should balance all factors to produce their desired sound. For example, too much downforce at a low speed will sound crunchy, whereas too little pressure will result in an unfocused sound as the bow slides across rather than grabbing the string. Similarly, if the bow is too close to the bridge with heavy downforce, it will make a boisterous sound. Advanced techniques like *legato*, *spicaato*, *louré*, or *staccato* are realised by adjusting bow speed, downforce and bow-bridge distance together [38], which are applied to present more diverse expressions. A particular piece of music can be performed in various timbres and volumes when a performer applies the bowing gestures differently. Thus the bowing parameters may help to describe the individual characteristics of a player. However, measuring bowing gestures from audio recordings is not directly feasible. One way to address this issue is to find relevant audio features representing the gestures. We thus design and extract timbre features validated for describing bowing gestures [39] to model players' bowing characteristics. The feature selection and calculation procedure are introduced in Section 2.4.4, and the violinist identification method using these features will be proposed in Section 4.2.

The more detailed definition of advanced violin playing techniques is beyond this Thesis's scope, but they can be found and investigated in many related books [40, 41].

2.4 Audio Features for Analysing Performer's playing Style

In this Thesis, the aim of developing audio features is to find descriptors of audio content relevant to the performer's style. These descriptors are further used to discriminate and identify violin performers. Therefore, the key to the performance of the violinist identification approach is the features' validity and the correlation between the features and the playing style. This section provides an overview of audio features previously and commonly used to analyse musical expressions and model performers' playing styles; the related feature extraction methods are also reviewed.

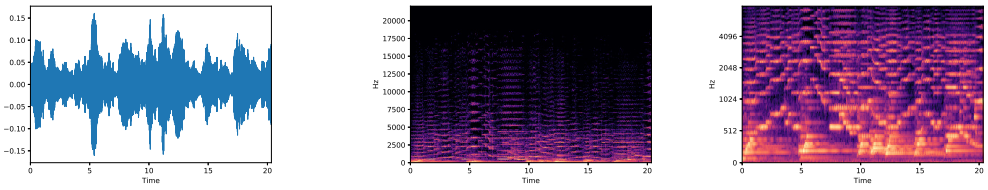
Audio features are generally classified based on the level of the information they describe. The *low-level* features reflect physical properties or musical factors related to the signal, such as the root mean square (RMS) energy and spectral centroid. The *mid-level* features generally summarise or present the low-level features in statistical methods, such as tempogram. The *high-level* features, sometimes also denoted as semantic features, reveal aspects that are usually close to how humans perceive music [42]. In this section, basic audio representation methods are firstly reviewed, followed by three kinds of features that are considered at high-level, including pitch features, timing features and timbre features.

2.4.1 Audio Representations

The common music signals produced by vibrating bodies (like strings on music instruments) are approximately periodic for short durations, which is also denoted as the repeat of sound pressure over time [43]. The intuitive way to capture audio signal is by taking samples of the air pressure over time, which is always presented as *waveform*. As Figure 2.5(a) shows, a waveform is one-dimensional sequential data which consists of audio samples that specify

the amplitudes at time-steps [44]. The sample rate can be varied, indicating how many samples per second (commonly $44.1KHz$, $22.5KHz$, or $16KHz$).

In addition to amplitude, frequency is another important attribute of an audio signal. The Fourier transform [45] is a mathematical function that decomposes a signal into its frequencies and the frequency's amplitude. To further find the relationship between time series and frequency, the Short-Time-Fourier-Transform (STFT) [46] is applied to provide a time-frequency representation. The extended audio is first divided into short segments of equal length, then computes the Fourier transform separately on each short audio segment to obtain the results of STFT. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. It is commonly used to visualise the STFT result of an audio signal. Figure 2.5(b) shows the spectrogram of the same audio that is demonstrated in Figure 2.5(a).



(a) waveform (b) spectrogram (c) Mel-spectrogram

Figure 2.5: The different representations of an audio signal.

However, the STFT does not match the frequency resolution of the human auditory system, so it is not the most popular choice in music analysis [44]. To address this problem, the Mel-spectrogram, which compresses the STFT in the frequency axis into the Mel-band, can effectively retain the most crucial perceptual information. There are many implementations of transferring the natural frequency to mel-band, [47] suggests the formula as Equation 2.1, where f denotes the original frequency and m means mel-frequency. An example of a Mel-spectrogram is shown in Figure 2.5(c).

$$m = 2595 \log 10(1 + \frac{f}{700}) \tag{2.1}$$

2.4.2 Pitch Features

All periodic tones (except sine wave) contain several frequency components called *Harmonics*. The lowest harmonic component is named as *fundamental frequency* (Hertz, [Hz]), also called *f0*, and the frequency of other harmonics are integer multiples of the fundamental frequency. For example, the fundamental frequency of A440 is 440Hz, and it has other harmonics at 880Hz, 1320Hz or 1760Hz. The subjective psychological dimension of the fundamental frequency is called *pitch*, and the pitch is associated with and quantified by fundamental frequency [48].

2.4.2.1 Pitch Estimation

Pitch is a critical perceptual attribute of music, an indicator of how "high" or "low" a melody is [48], and has a significant impact on the listener's perception of music [49]. Earlier works focused on pitch estimation on a single monophonic source, such as YIN [50] and PYIN [51] algorithm. Later works extract the main melody or multiple pitches from polyphonic music [52, 53]. Recently, most approaches have used deep neural networks for pitch tracking. For example, CREPE [54] is a monophonic pitch tracker built on a convolutional recurrent neural network, outperforming the previous state-of-the-art system. Rigaud and Radenen proposed a Bidirectional LSTM model to extract melody from singing voice [55], where a multi-scale convolutional architecture with a "harmonic" loss function is applied in this approach. For multiple f0 estimation from polyphonic music, a multi-task learning method is proposed and performed good results for various music sources [56].

The pitch estimation methods can be further applied for music transcription [57, 58, 59], music expression analysis [60], and music perception anal-

ysis [49]. However, it has been validated that compared with the individual pitch of single notes, listeners are more sensitive to the pitch variances and pitch tendency [49]. Therefore, features that reflect pitch variances attracted much attention, such as pitch histogram [61] and vibrato. Since vibrato is frequently applied in violin playing and is used in this Thesis to model the violinist's style, we will introduce the vibrato features in the following sub-section.

2.4.2.2 Vibrato Features

The vibrato plays an important role in the performance of vocal, flute and bowed string instruments, where it is often used to enhance selected notes and make them more prominent [62].

Generally caused by a slight oscillation in pitch or volume, vibrato makes a long sustained note more lively and energetic. For wind and brass instruments, vibrato is produced by altering the outward flow of air, which results in a small periodic change in volume [63]. For stringed instruments, vibrato is created by changing the length of the string, which is done by moving the finger back and forth. As this Thesis focuses on modelling the characteristic of violin playing, the latter vibrato definition is more suitable in this case. Assuming that the amount of pitch variation (defined as the "extent of vibrato") and the speed of pitch variation (defined as the "rate of vibrato") can represent a player's finger movement habits when playing a vibrato, these two metrics are then used to model a violinist's vibrato playing characteristics. Based on this assumption, we design and extract vibrato features and use them to identify violinists, the details of which will be presented in Section 4.2.

In previous work, a popular method of estimating vibrato parameters is to obtain the rate and extent from the location and values of the peaks and troughs in the fundamental frequency trajectory [64, 65, 66, 67]. Another

vibrato analysis method is to consider the fundamental frequency curve as a time-domain signal and computes its Fourier transform to estimate the vibrato parameters in the frequency domain [68, 69, 70]. In this Thesis, to intuitively understand the vibrato characteristics among performers, we extract vibrato features based on the pitch curve of music notes, which will be presented in Section 4.2.1.

2.4.3 Timing Features

The general music timing is described as human capabilities for processing temporal patterns [43, 71]. In particular, the timing presents the temporal events and their organisations in music [42]. Most relevant descriptors are extracted based on locating the note events, typically the note onsets. Higher level rhythmic features, such as the beat and metrical structure, can be obtained by measuring the periodicity of note onsets [72]. Figure 2.6 shows a waveform of a music clip, as well as locations of note onsets and beat.

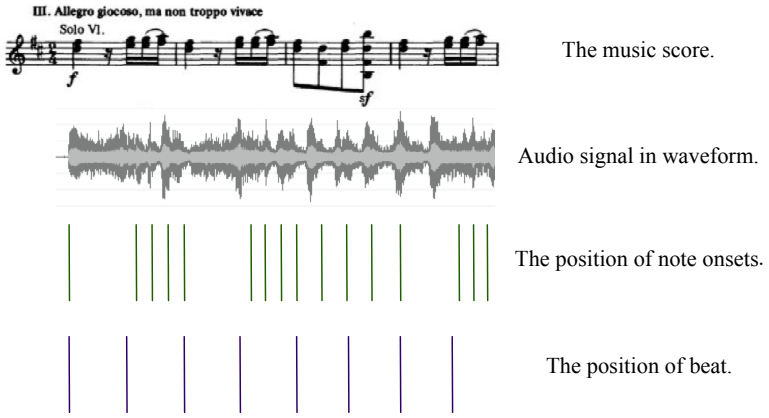


Figure 2.6: The location of beat and note onsets in a music clip.

Previous researches reveal that the timing features (e.g., duration of a note or beat, tempo, and onset interval times) are greatly related to the musical expression, music emotion cognition and performers' playing characteristics [43, 73, 74]. It is therefore assumed that the timing features can describe

the playing style of the violinist and thus can be used to distinguish violinists. Although many researches developed automatic onset detection (AOD) methods [75, 76, 77, 78, 79] and beat tracking algorithms [80, 81, 82], they are not a straightforward procedure on violin audio. The glissando, bow direction change and other issues frequently occur in violin performances would bring noises to the violin onset detection results [83]. Therefore, we manually annotate the note onset times and extract note-level timing features. Section 3.2.3 will describe the data annotation strategy, and Section 6.2 will introduce the timing feature extraction methods .

2.4.4 Timbre Features

The music's timbre is known as tone colour or perceived tone quality, which distinguishes different types of sound production. It enables listeners to identify musical instruments or recognise different singers or speakers. In signal processing, timbre is described as a multi-dimensional attribute, mainly determined by the frequency content of the sound in its time history [84, 85, 86]. Thus the spectral features of audio are widely used for analysing music timbres in MIR studies.

In violin performance, due to different players have their preferred bowing habits, the application of bowing gestures becomes a key factor in distinguishing violinists. However, measuring bowing gestures from audio signals is not feasible. Previous research attempted to establish the relationship between audio features and bowing gestures, so these features can be used to measure bowing parameters. Several timbre features including brightness, root-mean-square energy (RMS), spectral centroid, spectral contrast, spectral flatness or skewness, have been strongly correlated with bowing gestures [87, 88, 39]. Therefore the timbre features are assumed as important cues to describe the performer's bowing characteristic in this Thesis. We extract six timbre features, including RMS, spectral centroid, spectral band-

width, spectral contrast, Mel-frequency cepstral coefficients (MFCCs) and Zero-crossing Rate (ZCR), to present the performer's style, which will be introduced separately. (Although RMS is not usually regarded as a timbre feature, it is correlated with both volume and bowing force [89] and has been used to characterise spectral and harmonic properties in MIR studies [90]. Therefore, we include RMS in the timbre feature set in this Thesis.)

- **Root Mean Squared Energy**. The energy envelope of an audio signal is calculated using root-mean-square (RMS) energy on a short time frame basis. The RMS is calculated using Equation 2.2, where T is the frame size, and a_t is the amplitude of t^{th} audio sample in the frame.

$$RMS = \sqrt{\frac{1}{T}\sum_{t=0}^{T-1}(a_t)^2} \tag{2.2}$$

- **Spectral Centroid**. Spectral centroid is a commonly used statistical feature computed from the spectra. It indicates the centre of "gravity" of the spectrum and is always connected with the "brightness" of the sound [91]. The individual centroid of a spectral frame is defined as the average frequency weighted by amplitudes, divided by the sum of the amplitudes. The value of spectral centroid of the i^{th} audio frame C_i is defined as:

$$C_i = \frac{\sum_{k=0}^{K-1} f_k * Y(k)}{\sum_{k=0}^{K-1} Y(k)} \tag{2.3}$$

  f_k is the centre 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.

- **Spectral Bandwidth**. The bandwidth of the spectrum is described by the spectral spread [92], which is the spectrum's second central moment. To compute the Bandwidth (B_i), one has to take the deviation of the spectrum from the *Spectral Centroid*, according to Equation 2.4.

$$B_i = \frac{\sum_{k=0}^{K-1}(f_k - C_i)^2 * Y(k)}{\sum_{k=0}^{K-1} Y(k)} \tag{2.4}$$

- **Spectral Contrast**. The Spectral contrast is defined as the decibel difference between peaks and valleys in the spectrum [93]. The Fast Fourier Transform (FFT) is first applied to get the spectral components of each frame and then divided into six octave-based sub-bands. Finally, spectral contrast is estimated from each octave sub-band [94].

- **Mel-frequency Cepstral Coefficients (MFCCs)**. The MFCCs of a signal are a small set of features that concisely describe a spectral envelope's overall shape. The input audio is first segmented into short frames, and the framed signal is converted to the frequency domain using the Fast Fourier transform (FFT). The magnitude spectrum of the transformed signal is passed through the Mel filterbank and then converted to a logarithmic scale. The Mel warping is to convert the frequency into a perceptually meaningful scale using the following equations: $M(f) = 2595log_{10}(1+f/700)$ or $M(f) = 1127ln(1+f/700)$ [95], where higher resolution is assigned to lower frequency components. Finally, the discrete cosine transform (DCT) [96] of the list of Mel log powers are calculated, and the MFCCs are the amplitudes of the resulting spectrum. The "log-DCT" analysis, also referred to as the "cepstral" analysis, is commonly used to decorrelate convolved data for feature representation [42]. After referring to related papers [97, 98, 99], and balancing the computational complexity and the spectral information of the features, the first 13 Mel-frequency cepstral coefficients are

computed and applied in this Thesis.

- **Zero-crossing Rate (ZCR)**. The zero-crossing rate indicates the number of times a signal crosses the zero axis [100]; it occurs when two successive samples have different signs. The discrete audio signal x is first divided into I frames where $\{x_i : 1 \leq i \leq I\}$. Then, for i^{th} frame, there are T samples in the frame, and the zero-crossing rate is defined as:

$$Z_i = \frac{1}{2T} \sum_{t=1}^{T} sign|[x_i(t-1) - x_i(t)]| \tag{2.5}$$

where

$$sign(v) = \begin{cases} 1, & v > 0 \\ 0, & otherwise \end{cases}$$

Since periodic sounds tend to have a small value of it, while noisy sounds have a high value of it, the ZCR is frequently used in speech/-music classification [101], instrument classification [102] and sounds identification [103].

2.5 Statistical Models and Music Similarity

In this Thesis, we assume that the global distribution of audio features from a performer's all performances can be used to model this performer's playing style. Various statistical models are applied and compared to investigate which model performs better in modelling violinist's styles. Section 2.5.1 outlines the definition and application of three statistical models used in this Thesis, including Histogram, Kernel Density Estimation (KDE) and Gaussian Mixture Models (GMM). Next, since we propose methods for violinist identification based on the calculation of musical similarity, Section 2.5.2

reviews related research in music similarity.

2.5.1 Statistical Model

The statistical model is a mathematical representation that embodies a set of statistical assumptions concerning the properties of sample data [104]. When data analysts apply various statistical models to their investigative data, the information can be understood and interpreted more strategically. Rather than observing the raw data, this practice allows them to identify relationships between variables (e.g. mean or variance of a series of variables), make predictions about future data sets, and visualise data intuitively. In music performance, it is difficult to reveal the relationship between the performer's style and audio features at note-level (or frame-level). Still, the feature distribution of hundreds or thousands of notes may indicate the performer's characteristics. This section will introduce the definition and calculation method of three statistical models, including Histogram, KDE and GMM; their applications in MIR research are also reviewed.

2.5.1.1 Histogram

Estimating *probability density function* (PDF) of a series of data points is a central topic in statistical research, which is also called *density estimation*. The histogram is perhaps the simplest way of density estimation and was firstly introduced by Karl Pearson [105]. It is an approximate representation of the numerical data's distribution and gives a rough sense of the underlying data distribution density.

The first step of constructing a histogram is to divide the entire range of data into a series of bins (or intervals), and the width of a bin is called "bin size". The bins are usually specified as consecutive, non-overlapping intervals of a variable, and they must be adjacent and are mostly in equal size [106]. The number of samples that fall into each interval is denoted as

"frequency", displayed as a rectangle erected above the bin, and the rectangle's height is proportional to the frequency. As the adjacent bins leave no gaps, the rectangles of a histogram touch each other to indicate that the original variable is continuous [107].

In MIR studies, the histogram is always used to obtain the audio feature distribution of a music piece, from which high-level properties of the music can be observed and captured. For example, pitch histograms depict the pitch distribution of a song, and they can be applied to the musical genre classification [61] and music repetition detection [108]. Moreover, the beat histogram is related to rhythmic similarity [109]; multi-dimensional histograms using multiple audio features are validated for music emotion recognition [108].

2.5.1.2 Kernel Density Estimation

The kernel density estimation uses a kernel to smooth frequencies over the bins, which can accurately reflect the distribution of an underlying variable. Compared with the histogram, the KDE is usually plotted as a curve rather than a set of bars and yields a smoother probability density function. The definition of KDE is shown in Equation 2.6,

$$\hat{p}(x) = \frac{1}{jh} \sum_{i=1}^{j} W\left(\frac{X_i - x}{h}\right) \tag{2.6}$$

where W denotes the kernel function that is generally a smooth, symmetric function (such as a Gaussian that is calculated using Equation 2.7), and $h > 0$ is a smoothing parameter called the bandwidth. The $(X_1, X_2, X_3, X_4, ...X_j)$ are observation data points, and x is a linearly spaced series of data points which houses the observation data points. The individual kernel value $W(x)$ is calculated using Equation 2.7 (if the kernel function is Gaussian). The KDE smoothes each data point X_i into a small density bump and then sums all these small bumps together to obtain the final density estimate result.

$$W(x) = \frac{1}{\sqrt{2\pi}} \exp(\frac{-x^2}{2}). \tag{2.7}$$

Due to its good performance in modelling and visualising data distribution, KDE is used for music classification [110], emotion recognition [111, 112], or music similarity analysis [113] in previous research.Therefore, we also applied KDE to obtain the audio feature distribution of performers, which is used to describe the performer's performance style, the details of which are presented in Section 4.2.

2.5.1.3 Gaussian Mixture Model

Gaussian distribution is a widely used statistical model in many research areas, which has a bell-shaped curve, with the data points symmetrically distributed around the mean value. The probability density function of a one dimensional Gaussian distribution $p(x)$ is given by Equation 2.8:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{2.8}$$

where σ is the mean and μ^2 is the variance. However, if a large data set has more than one "peak" in its distribution, trying to fit it with a unimodal model will usually result in a poor fit. An obvious way to model a multimodal distribution is to assume it is generated from multiple unimodal distributions. Motivated by this assumption, it is reasonable to model multimodal data as a mixture of many unimodal Gaussian distributions, and the "Gaussian mixture model" is then presented.

A Gaussian mixture model is parameterised by the weights of the mixture component and the means and variances/covariances of each component. For a Gaussian mixture model with C components, the c^{th} component has a mean value μ_c and variance value σ_c for the univariate case and a mean

vector $\vec{\mu}_c$ and covariance matrix Σ_c for the multivariate case. The mixture component weights are defined as ϕ_c for component c with the constraint that $\sum_{c=1}^{C} \phi_c = 1$, so that the total probability distribution normalises to 1. For the univariate case, the distribution $p(x)$ is:

$$p(x) = \sum_{c=1}^{C} \phi_c N(x|\mu_c, \sigma_c) \tag{2.9}$$

with

$$N(x|\mu_c) = \frac{1}{\sigma_c\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_c)^2}{2\sigma_c^2}\right) \tag{2.10}$$

If each component is a multi-variant gaussian function, and each component density is a D-variate Gaussian function of the form, the probability distribution function is described as:

$$p(\vec{x}) = \sum_{c=1}^{C} \phi_c N(\vec{x}|\vec{\mu_c}, \Sigma_c) \tag{2.11}$$

with

$$N(\vec{x}|\vec{\mu_c}, \Sigma_c) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_c|}} exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu_c})^T\Sigma_c^{-1}(\vec{x}-\vec{\mu_c})\right\} \tag{2.12}$$

The GMM is mostly used for voice recognition and speaker recognition [114, 115, 116] with good performances. Particularly, in MIR research, the GMM is applied for calculating music similarities [117], classifying music genres [118] and recognising instruments [119, 120]. Thus we hypothesise the GMM would help identify and model the style of instrumentalists, details of which will be shown in Section 4.2.

2.5.2 Music Similarity Analysis

Although there is little research using musical similarity methods to identify performers, prior work in related fields provides useful insight into this Thesis's topic. Music similarity is a broad area that has been studied for a long time, and it has been applied to many MIR studies, such as playlist generation [121], music recommendation [122], emotion recognition [123], music genre classification [124] and instrument classification [125]. We roughly summarise the three steps in assessing similarities between musical pieces. First, it is assumed that some aspects of the audio signal are kept time-invariant over a short period. Thus most audio features are designed and extracted from a short time window (also called a "frame") of the audio signal. Next, since audio features at the frame level are not representative of structural or global musical characteristics, we need to analyse the audio features statistically. If the feature distribution is known, it is straightforward to calculate statistical parameters such as the mean and variance. If the feature distribution is unknown or complex, it is necessary to develop more sophisticated statistical models with parameters learned or trained from the data [32]. Finally, the distance or divergence between feature distributions can be calculated as an indicator of musical similarity estimation.

Previous music similarity calculation systems rely on statistical models that are fitted on low, medium or high-level audio features [126]. For example, MFCCs and histograms are applied to model timbre characteristics, and Earth Movers Distance [127] is computed to measure music similarities. Similarly, a GMM trained on spectral coefficients of each song has been shown to represent musical characteristics well. The Kullback-Leibler (KL) divergence is then calculated to evaluate music similarity between Gaussian distributions [128].

The distance between feature distributions is a key parameter to represent music similarity. KL divergence [129] is a common way to measure how one

probability distribution is different from another, and the JS divergence [130] is a symmetric version of the KL divergence. In terms of discrete probability distributions, P and Q are defined on the same probability space, while P represents the distribution of observation data, and Q represents a model, or an approximation of P. Then the KL divergence can be calculated as:

$$D_{KL}(P \,||\, Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{2.13}$$

KL divergence is asymptotic to infinity when one of the distributions tends to zero, which also allows it to be used successfully in many scenarios [131]. Particularly, since the KL divergence between GMMs is not analytically tractable, we use matching-based approximation following the implementation in [132] in this Thesis. The KL-divergence between the Gaussians $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$ is defined as follows, and Tr means the trace of matrix.

$$D_{KL} = \frac{1}{2}(log\frac{|\Sigma_2|}{|\Sigma_1|} + Tr(\Sigma_2^{-1}\Sigma_1) + (\vec{\mu_1} - \vec{\mu_2})^T\Sigma_2^{-1}(\vec{\mu_1} - \vec{\mu_2})) \tag{2.14}$$

2.6 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows algorithms to use historical data (also known as *training data*) as input to predict new output values accurately. It is widely used for both classification and regression tasks.

When training a machine learning model, the training data is a set of data samples that are denoted as $X = (X_1, X_2, ..., X_j)$ and $X_j \in \mathbb{R}^n$ where n means the dimension of each data sample, and j means the number of training samples. The corresponding output labels are denoted as $Y = (Y_1, Y_2, ..., Y_j)$. The machine learning is generally categorised as *supervised Learning*, *unsu-*

pervised learning and *reinforcement learning*. When a model is trained with supervised learning, the data sample X and the corresponding label Y are considered as input and the resulting model can predict $\hat{Y}$ from the new data point $\hat{X}$. For unsupervised learning, on the other hand, the training data contains only samples without any target labels, and the goal is to find hidden patterns or relationships among such data by clustering or density estimation. As for reinforcement learning, the algorithm needs to interact with a dynamic environment and achieve a certain goal, such as driving a car or playing a specific game.

In this Thesis, we only use supervised learning to identify violinists. Since two machine learning models, including the K-Nearest Neighbour (KNN) and the Support Vector Machine (SVM), are considered as baselines in Chapter 4, their definitions and computations are first presented in this Section. In addition, we will propose a transfer learning method for violinist recognition (see Chapter 7), where two categories of deep learning models are used, including convolutional neural network (CNN) and convolutional recurrent neural network (CRNN). We present the basic definitions of the two models in Section 2.6.2, together with an overview of the transfer learning approach. Other classification algorithms, such as Decision Trees and Hidden Markov Models, are not included here due to this Thesis's scope.

2.6.1 Machine Learning Models

2.6.1.1 K-Nearest Neighbours

The K-Nearest Neighbours algorithm (KNN) is a supervised learning method which was first developed by Evelyn Fix and Joseph Hodges [133] and later expanded by Thomas Cover [134]. It is used for both classification and regression tasks. The input is K closest training examples in a dataset, and the output is a class label for the classification task; or the value of an object's

property for the regression task.

As we discussed earlier, the training data are $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)...$ (X_j, Y_j), each sample contains a multidimensional feature vector X_j with a label Y_j. The training phase of the KNN is implemented by storing the feature vectors and labels of all training samples, and K is a constant that users initialise. In the test phase, unlabeled feature vectors $\hat{X}$ are set as test data. For each $\hat{X}_i \in \hat{X}$, the distance between $\hat{X}_i$ and each training data sample $X_i \in X$ is firstly calculated, and the nearest K training samples from the $\hat{X}_i$ can obtain. Finally, for classification tasks, the most frequent class label in such K training samples is assigned as the predicted label $\hat{Y}$ for the $\hat{X}$; for regression tasks, the mean of the K labels is returned as the predicted value $\hat{Y}$.

Specifically, the distance between test data and each training data is mostly measured by Euclidean distance [135]. For *n-dimensional* data vectors p and q given by Cartesian coordinates in Euclidean space, the distance $d(p, q)$ between p and q is defined by:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2} \tag{2.15}$$

2.6.1.2 Support Vector Machine

Support Vector Machine (SVM) is another popular supervised learning model for solving pattern recognition problems. It was introduced and developed by Vapnik et al. [136, 72, 137], and then widely used for multi-class classification and regression tasks.

In the training phase of SVM, the algorithm is performed by finding a hyperplane in an n-dimensional space (n means the dimension of each input sample vector) that separates the training data in different classes distinctly.

Although many possible hyperplanes could be selected to classify data, our objective is to find a plane with the maximum distance among data points of any class. The distance is known as *margin*, so the SVM is also named as *Maximum Margin Classiffier*.

If the data is linearly separable in the SVM classifier, we use a linear kernel function to fit a linear hyperplane between two classes. The margin is also called hard-margin [137]. If the data is non-linear separable, the soft-margin method [137] and kernel trick [138] method are introduced to fit a non-linear boundary between classes. Details of the mathematical process of SVM are beyond this Thesis's scope, but they can be checked in [139]. In MIR research, the SVM is frequently used as a classifier to achieve emotion recognition [140], music genre classification [141], and instrument classification [142]. In particular, it is also used for performer identification and works well [143, 144]. Therefore, SVM is used as a baseline to evaluate the performance of our proposed method, details of which will be introduced in Chapter 4.

2.6.2 Deep Neural Networks

Although audio features and statistical models (or machine learning methods) can be used for music classification or performer identification, they require appropriate feature design to present audio properties, which depends on domain knowledge and careful engineering work. To automatically discover the needed representations to model and classify music expression, deep neural networks (DNN) have become very popular in MIR research (e.g. music classification, transcription, music generation) [145, 146, 147] in recent years. This trend is even stronger in other fields, such as natural language processing and computer vision. In this Thesis, to improve the performance of violinist identification using a limited dataset, we will apply the transfer learning method in Chapter 7. This section introduces a basic definition

of deep learning architectures, including Convolutional Neural Network and Convolutional Recurrent Neural Network, followed by an overview of the transfer learning approach.

2.6.2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a class of deep learning architectures commonly applied to analyse visual imagery [148]. CNN assumes features in different hierarchical levels and can be extracted by convolutional kernels. The hierarchical features are learned to achieve a high-level task during supervised training [145]. A standard CNN consists of an input, hidden, and output layer. In a convolutional neural network, the "convolution" is typically performed in hidden layers, which implement a dot product of the convolution kernel with the layer's input feature matrix. This product is usually calculated by Frobenius inner product [149], and its activation function is commonly Rectified Linear Unit (ReLU), which can be formulated using $g(z) = max(0, z)$. As the convolution kernel slides along the input matrix, the convolution operation generates a feature map, contributing to the next layer's input. This is followed by other layers such as pooling layers, fully connected layers, and normalisation layers [150].

Since CNN was originally designed for image inputs, mostly two-dimensional (2D) signals per channel, it is difficult to apply CNN directly to audio signal (one-dimensional sequential data) in early studies. To solve this problem, the audio signal was converted to a 2D representation and the representation was regarded as a visual image that contained audio characteristic information. Then CNN could work well in this case. The two dimensions are mostly frequency and time, popular representations like Constant-Q transform (CQT) [151], Spectrogram [152], Mel-spectrogram [153], and Chromagram [154] were explored and compared as inputs for training CNN [155]. The CNN uses these time-frequency representations to play a powerful role

in tasks like music instrument recognition [156], music tagging [157], onset detection [158] or fundamental frequency estimation [159, 160].

2.6.2.2 Convolutional Recurrent Neural Network

RNNs are a class of deep neural architectures which are often used to model sequential data (e.g. audio signals or word sequences). They can use their internal state (memory) to process variable length sequences of inputs [161], which allows them to exhibit temporal dynamic behaviour. Hence the RNNs are applicable to tasks such as handwriting recognition [162], machine translation [163] or speech recognition [164]. The more detailed knowledge of RNNs is beyond this Thesis's scope, but their fundamental principles and application in MIR can be found in [165, 44].

A Convolutional Recurrent Neural Network (CRNN) is described as a modified CNN by replacing the last convolutional layers with an RNN. The CNN is a feature extractor, and RNN is used for temporal or structural information summariser [145]. This architecture was first proposed in [166] for document modelling and classification, and later applied for image classification [167], music transcription [59] and music classification [145]. Recently, the CRNN has been further modified for music tagging and singer identification (SID) with competitive results. For example, Minz Won et al. [168] proposed a self-attention-based deep sequence model for music tagging, where stacked transformer encoders follow the shallow convolutional layers. In this Thesis, the pre-trained CNN and CRNN models are considered as source tasks, and they are fine-tuned using a violin dataset to identify violinists (see Chapter 7). This technique is also known as transfer learning, which will be introduced in the following subsection.

2.6.2.3 Transfer Learning

As we discussed above, deep learning models such as CNN, RNN or CRNN perform powerfully for many tasks. However, a huge amount of labelled data are required in the training phase of DNN, which is time-consuming for data collection and labelling. Meanwhile, to ensure the performance of DNN, the feature space and distribution of test data should be the same as the training data, which leads to poor performance of a pre-trained model working on other tasks. To solve these problems, researchers assume that reusing knowledge from a task trained on a large dataset would help with a different but related new study. The performance of new studies would be maintained by tuning the pre-trained model. According to this assumption, transfer learning methods have been proposed and developed, and they have been highly successful in many fields [169, 170, 171]. It aims at improving the performance of target models using a small dataset by transferring the knowledge learned from source domains [172]. Tasks trained in the source domain in the past are generally referred to as *source task*, while new tasks are referred to as *target task*.

In CNN or other deep learning methods, the different levels of features are hierarchically learned from early to later layers. Therefore, the basic idea of transfer learning is that the learned low-level feature extractor can be transferred to target tasks in the source task. For example, in the computer vision field, the rich basic visual information such as the basic shapes or prototypical templates of objects were captured when trained for image classification. The learned knowledge can be transferred for target tasks like person re-identification [173] or object detection [174]. In MIR research, when training a model for music tagging, low-level information such as tempo, pitch, (local) harmony or envelope can be captured in early layers [157, 175], which can be transferred for other related tasks like music genre classification, emotion prediction, or audio event classification [176].

In this paper, due to our limited data, we propose a transfer learning approach to identify violinists. To the best of our knowledge, there is no publicly available dataset with the violinist's ID and no pre-trained models for instrumentalist classification. We therefore consider the neural networks trained for other related tasks as source tasks, then transfer the learned weights and fine-tune the models using our datasets to identify violinists. Details of these methods will be presented in Chapter 7.

2.7 Evaluation Methods

To evaluate the performance of proposed violinist identification methods, we use *F-score* and *Confusion Matrix* that have been universally used to assess multi-class classification tasks. In addition, to reduce the bias caused by splitting the dataset in a simple way, we apply cross-validation in this Thesis. This section introduces the definition of mentioned evaluation metrics and the process of cross-validation.

2.7.1 F-score

The *F-score*, also called the *F-measure*, is a evaluation metric of a model's performance. It is originally designed to evaluate binary classification systems, which classify examples into *"positive"* or *"negative"*. The *negative class label* and *positive class label* are respectively represented as *"0"* and *"1"*. For each test point, the ground-truth label and the predicted label are denoted as (Y_j) and $(\hat{Y}_j)$. The result is evaluated using precision (P),

recall (R) and F1-score (F_1). They are defined as:

$$P = \frac{O_{tp}}{O_{tp} + O_{fp}} \tag{2.16}$$

$$R = \frac{O_{tp}}{O_{tp} + O_{fn}} \tag{2.17}$$

$$F_1 = 2 * \frac{P_1 * R_1}{P_1 + R_1} \tag{2.18}$$

where O_{tp}, O_{fp}, O_{fn} and O_{tn} are the numbers of *true positives* (TP), *false positives* (FP), *false negatives* (FN) and *true negatives* (FN), respectively. *TP* means a test point is correctly predicted as the positive class i.e., $Y_j = \hat{Y}_j = 1$. Similarly, *TN* means a test point is correctly predicted as the negative class i.e., $Y_j = \hat{Y}_j = 0$. *FP* means a test point is incorrectly classified in the positive class, i.e., $Y_j = 0$ while $\hat{Y}_j = 1$. Additionally, *FN* means a test point is incorrectly predicted as the negative class, i.e., $Y_j = 1$ while $\hat{Y}_j = 0$.

However, in multi-class classification tasks, the standard *F-score* is unsuitable. To solve this problem, the *macro F-score* and *micro F-score* are introduced [177] and frequently used in many relevant fields. The *macro F-score* is defined in following equations, where P_z and R_z denote the *precision* and *recall* of z^{th} class, Z means the number of classes. The *Macro Precision* and *Macro Recall* are computed by taking the average of the precision and recall of the system on different classes, and the F_{macro} is simply obtained by calculating the harmonic mean of these two parameters.

$$P_{macro} = \frac{1}{Z} \sum_{z}^{Z} P_z \tag{2.19}$$

$$R_{macro} = \frac{1}{Z} \sum_{z}^{Z} R_z \tag{2.20}$$

$$F_{macro} = 2 * \frac{P_{macro} * R_{macro}}{P_{macro} + R_{macro}} \tag{2.21}$$

In Micro-average method, the individual *TP*, *FP*, *FN* and *TN* of the system are summed up for different classes and then applied to get the statistics. The *micro F-score* is calculated using equations below.

$$P_{micro} = \frac{\sum_z^Z O_{tp}^z}{\sum_z^Z O_{tp}^z + \sum_z^Z O_{fp}^z} \tag{2.22}$$

$$R_{micro} = \frac{\sum_z^Z O_{tp}^z}{\sum_z^Z O_{tp}^z + \sum_z^Z O_{fn}^z} \tag{2.23}$$

$$F_{micro} = 2 * \frac{P_{micro} * R_{micro}}{P_{micro} + R_{micro}} \tag{2.24}$$

In sum, the Macro-average method can be used to measure how the system performs across the sets of data, and the micro-average can be helpful when the dataset varies in size for a different class. In this Thesis, the performance of the whole system across the data from each performer is more important, and the amount of data for each performer is equivalent. Therefore, in our case, the macro-F1 is more suitable, and it is applied in the following Chapters to evaluate the performances of our proposed methods.

2.7.2 Confusion Matrix

Although the macro-F1 can evaluate the performance of a classification model, it hides the detailed results for each label. For example, with three or more classes in the dataset and we obtain a macro-F1 of 80%, it is uncertain if all classes are being predicted equally well, or if one or two categories are being neglected by the model. Therefore the confusion matrix is introduced to solve this problem and has been widely used in many existing works [15, 176].

In a multi-classification task, if there are Z classes in the dataset, the confusion matrix is a $Z * Z$-dimensional matrix, with each row corresponding to a predicted class and each column corresponding to an actual class. The diagonal elements represent the number of samples where the predicted labels

are equal to the true labels. In contrast, the non-diagonal elements are those samples mislabelled by the classifier. A higher diagonal value of the confusion matrix indicates many correct predictions. In addition, the confusion matrix can be visualised using the heat map function, where the shades of colour can also mean the model's performance.

In our case, we expect a violinist identification approach that works well for recognising each performer in the dataset. Therefore, we apply the confusion matrix in this Thesis to evaluate the method's performance. An example of a confusion matrix can be found in Section 4.3.2.4.

2.7.3 Cross-Validation

The dataset is usually split in machine learning experiments into a training set and a test set, where the model is fitted on the training set and evaluated on the test set. If we split the data in a simple way, the model could be biased to the characteristics of the training set, resulting in overfitting or selection bias problems [178]. To address this issue, cross-validation has been proposed and is widely considered a better way to evaluate the performance of models.

The cross-validation is also called rotation estimation, and the general procedure is shown as follows:

1. Shuffle the dataset randomly;
2. Split the dataset into k groups;
3. For each unique group of data:
    1) Take the group as test data set;
    2) Take the remaining groups as a training data set;
    3) Fit a model on the training set and evaluate it on the test set;
    4) Retain the evaluation score and discard the model;

4. Average evaluation metrics are computed in the loop to obtain the model performance.

There are many types of cross-validation like *Leave one/P out cross-validation*, *Leave one/P group(s) out cross-validation* [179]. This Thesis uses *Leave one group out cross-validation(LOGOCV)* to evaluate the performance of the violinist identification system, where the dataset is split into groups according to which music piece they are associated with.

2.8 Summary

This Chapter outlined the technical background of this Thesis. The music expression and its influential factors are first presented in Section 2.2, where the importance of performers' interpretation of given music is also discussed. Since this Thesis uses violinist identification as a case study for performers' style modelling, the structure of the violin and fundamental playing techniques are introduced in Section 2.3. Next, we reviewed the definitions and applications of audio features, statistical models and music similarity algorithms, which can be used to describe, model and measure music characteristics. In Chapter 4, 5, and 6, such techniques will be applied to model violinists' characteristic and identify violinists. Finally, Section 2.6 and Section 2.7 introduced the machine learning methods and corresponding evaluation metrics, which will be applied to characterise violinists and assess the performance of the proposed methods.

In the following Chapters, we will propose violinist identification methods applicable to different scenarios. However, before elaborating on these methods, we need to build a set of violin datasets containing violinist labels to evaluate these methods' effectiveness. We will introduce the details of the dataset construction procedure in Chapter 3.

Chapter 3

Dataset Construction

3.1 Introduction

Violinist identification (VID) takes an audio signal as input and the ID or name of the performer as output. A violin dataset containing multiple performer IDs is intuitively needed if we want to evaluate a VID approach. However, most existing open-access violin datasets have been built for expression classification [21], instrument identification [180] or pitch estimation [181], and these datasets do not contain the label of performer's name (or ID). To evaluate our approach presented in the following sections, we constructed two groups of datasets, which are annotated from commercial concerto recordings and solo scale recordings, respectively.

As the characteristic playing styles developed by virtuoso violinists are favoured by most listeners and often imitated by many students, it is vital to study how these styles are developed and which acoustic features can be used to describe them. Therefore, three datasets are created from recordings of commercial concertos performed by nine virtuoso violinists, which are applied for different VID scenarios. Details of the recording selection and dataset creation methods are presented in Section 3.2.

However, the selected concertos are recorded in different environments, which may introduce noises in modelling violinist's playing styles. Suppose we use timbre features to model a violinist's style. In that case, such features may not only reflect the violinist's playing but also contain characteristics of the orchestra accompaniment or the property of the instrument. To solve this problem and validate the robustness of our methods, we construct another group of datasets from solo violin recordings. Basic information about the solo violin recordings is briefly described in Section 3.3.1, and the dataset construction methods are subsequently presented in Section 3.3.2 and Section 3.3.3.

3.2 Concerto Dataset Construction

3.2.1 Concerto Recording

Concerto is a musical work that focuses on a solo instrument, such as violin or piano, and is accompanied by an orchestra. It is paramount in the repertoire of master violinists who perform concertos individually. In violin performance, tempo, intensity and vibrato can vary from person to person since each violinist brings a personal style to their playing.

In addition, most concertos contain a cadenza part. Performers can play this without concern the coordination with the orchestra or obeying the global tempo. Violinists therefore often exhibit their unique playing style most expressively during the cadenza. Paying particular attention to the cadenza is thus very useful for our research to understand how to model differences in individual playing styles. In addition, we do not have to address the influence of accompaniment and can focus on features extracted from the solo performance.

We select five concertos written by five well-known composers: Beethoven, Brahms, Mendelssohn, Tchaikovsky and Sibelius. These pieces have all been

performed by nine violinists: Jascha Heifetz, Anne Sophie Mutter, David Oistrakh, Itzhak Perlman, Pinchas Zukerman, Isaac Stern, Salvatore Accardo, Yehudi Menuhin and Maxim Vengerov, who are all leading master violin players. All concerto repertoires are from commercial CD albums. Further details of repertoires are listed in Table 3.1, while information about the CD album is listed in the Appendix A. However, these concerto compositions were recorded by various performers at separate times, and differences in recording dates and conditions can introduce noise into the dataset construction. To address these issues, we conduct data pre-processing in our proposed violinist identification approaches, and try to select audio features (e.g., vibrato, expressive timing) that are less disturbed by the recording conditions.

Three datasets were constructed based on such concerto recordings to evaluate proposed VID approaches in different scenarios. The first dataset only contains isolated vibrato notes, and it will be used in Chapter 4 to assess the effectiveness of vibrato features in identifying violinists. The second dataset consists of selected short musical clips with onset annotations, which will be applied in Chapter 6. The third dataset is constructed from all concerto clips that contain violinists' performances, which will be used for research in Chapter 7. Each dataset's audio selection and annotation strategy will be separately presented in this section.

3.2.2 Isolated Vibrato Notes Dataset

To quantitatively investigate the importance of vibrato on a violinist's characteristic style, we annotate a certain number of vibrato notes from each performer's performance to build the Isolated Vibrato Notes (IVN) Dataset. Since violin performance is always accompanied by orchestra in concertos, to reduce the influence of accompaniment on vibrato features, we segment solo notes containing vibrato or vibrato notes with very low accompaniment

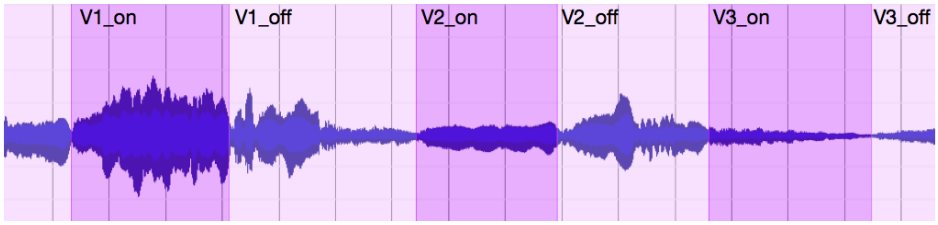


Figure 3.1: Vibrato notes segmentation (excerpt)

volume from original concerto recordings. We also sidestep the influence of variation between music pieces, the same excerpts from each concerto for every performer are thus segmented. This way, we can focus on the differences in vibrato characteristics among performers.

Although there are automatic vibrato detectors [182, 183] to locate the position of vibrato in a music piece, their accuracy for polyphonic violin recordings (especially concertos) has not been validated yet.

While automatic vibrato detectors can be used to locate vibrato positions in a music piece [182, 183], their accuracy for polyphonic violin recordings (especially concerto data) has not been verified yet. Since we need to ensure that vibrato features are extracted from vibrato notes, each vibrato note's onset and offset position must be accurately calibrated. Therefore, we manually label the onset and offset time of each vibrato note, rather than using automatic vibrato detectors. Sonic Visualiser [184] is used for data annotation, together with the Match Vamp plugin [185] to align the performances, guiding and improving the annotation performance. Figure 3.1 shows an example of the interface used for annotation and an excerpt of the data with several notes. Darker (purple) segments correspond to solo vibrato notes in this plot. Vibrato note onset and offset times are shown as dark purple vertical lines around segment boundaries.

The recordings and the amount of annotated data in each movement are listed in Table 3.1. The total amount of vibrato annotations for each performer is 248.

Table 3.1: Concerto vibrato note dataset. We annotated the vibrato note segments from the original recordings, '# annotations' refers to the number of vibrato note annotations in each movement.

Composer	Concerto Name	Movement	# annotations
L. V. Beethoven	Violin Concerto in D major, Op.61	I	21
		II	26
		III	4
J. Brahms	Violin Concerto in D major, Op.77	I	11
		II	6
		III	4
F. Mendelssohn	Violin Concerto in E minor, Op.64	I	13
		II	47
		III	3
P. I. Tchaikovsky	Violin Concerto in D major, Op.35	I	26
		II	7
		III	17
J. Sibelius	Violin Concerto in D minor, Op.47	I	23
		II	24
		III	16

3.2.3 Selected Concerto Clips Dataset

To identify the violinist from short music clips based on note-level features, we divide the original recordings into several clips and then select some clips to label the boundaries of each note to construct the Selected Concerto Clips (SCC) Dataset. The selection of concertos for this dataset is the same as in the IVN dataset.

Since we want to analyse the audio features at the note level, the onset position of each note must be precisely labelled. Although there are many existing automatic onset detectors [156, 186], their accuracy in violin performances has not been validated yet, thus they cannot be directly used for our purposes. We manually label onset times using Sonic Visualiser, and the overall data annotation procedure consists of three steps: music piece selection, alignment, and onset time labelling.

The first step is music piece segmentation and selection. In concerto performance, the violinist is not playing in the "Preludes" or "Interludes" (that are performed by orchestra alone). In addition, the player's performances

are drowned out by the orchestra in some parts. Therefore we cut out the parts of the music without the violin or where the violin cannot be heard clearly so that the original concerto recordings are divided into several short pieces. However, it is unfeasible to label all notes in five concertos manually. We thus select some typical clips by considering the impact of the pieces and note types. First, the tempo may be very different in different parts of a concerto movement. For example, the start of a movement is always soft and slow, while the middle part is more varied, and the ending is usually passionate. We assume performers use their preferable speeds when they play music with a different tempo, which can be used as a feature to model and classify performers. Second, different note types, such as semibreve, minim, crotchet, quaver, dotted note, etc., also affect performers' timing expression. Therefore, we selected at least three different parts with different speeds and covered as many note types as possible from each movement to ensure the diversity of the data.

In the second step, to make the feature extraction easier, we align the selected music pieces and the start of the first note is considered as 0 seconds for each chosen clip. Therefore the deviation of note onsets in selected pieces can be computed, and the onset feature extraction method will be proposed in Section 6.2.1.1.

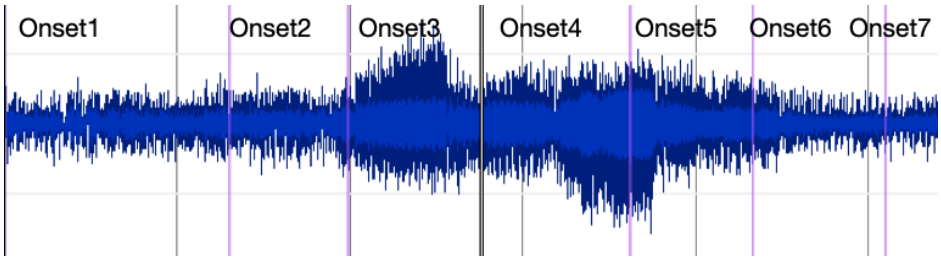


Figure 3.2: Note onset time annotations (excerpt)

The final step is onset time labelling. In practice, labelling short notes with correct onset times is a substantial challenge. For example, it isn't easy to label performance with a quick succession of short sixteenth notes. To

solve this, we slow down the music using the appropriate function provided in Sonic Visualiser [184]. Figure 3.2 shows the interface used for onset time annotation. The vertical line indicates the position of onset times. The number of annotated onset times from each movement is listed in Table 3.2.

Table 3.2: Concerto note segmentation dataset. We first cut the original recordings into several clips. We then select two or three clips from each movement and annotate the note onset time.' annotations' refers to the number of note annotations in each movement.

Composer	Concerto Name	Movement	Onsets
L. V. Beethoven	Violin Concerto in D major, Op.61	I	664
		II	239
		III	352
J. Brahms	Violin Concerto in D major, Op.77	I	262
		II	157
		III	193
F. Mendelssohn	Violin Concerto in E minor, Op.64	I	204
		II	201
		III	235
P. I. Tchaikovsky	Violin Concerto in D major, Op.35	I	225
		II	177
		III	148
J. Sibelius	Violin Concerto in D minor, Op.47	I	233
		II	200
		III	186

3.2.4 All Concerto Clips Dataset

To ease the workload of manual data annotation in the above section, we selected a minimum of three segments from each movement to annotate the note onset times. However, when training a deep learning model, a larger dataset can help improve its accuracy and generalisation [187]. Furthermore, if the input to the deep learning model were not musical notes, the onset label of each note would not be needed (Chapter 7 will present information about deep learning models). Thus, after removing the parts of the concerto without the violin or where the violin cannot be heard clearly, the remaining clips are all retained, forming the All Concerto Clips (ACC) Dataset, elim-

inating the need for music clip selection and onset time annotation. Each performer's performance in this dataset is approximately two hours.

3.3 Solo Dataset Construction

To address the issues raised in Section 3.1, we find recordings of violin solos played by 22 professional performers on 13 violins. We then annotate these recordings according to our research requirements and construct two datasets. The introduction of the solo recordings and the annotation method of each dataset are presented in this section.

3.3.1 Violin Solo Recording

During the European Bilbao project[1], 13 new (white) violins were designed and built and then evaluated within a free categorisation task by 22 professional violinists [188]. All violinists were invited to play a musical scale on each violin, and the musical scale contains around 37 notes. Moreover, all performers finished the recording task in the same studio (a large rehearsal room at the Bilbao conservatory) with the same equipment, and the distance and angle between each and the microphone remained as identical as possible. Under these circumstances, the recording condition and environment are well controlled. The differences in timbre or other aspects of violin performance are mainly affected by the instruments and the performer's interpretation.

3.3.2 Selected Solo Clips Dataset

Since we analyse the violinist's playing characteristics using note-level audio features in the following chapters, the note onset times need to be annotated precisely. However, to fairly compare the performance of a VID algorithm

[1] https://www.bele.es/en/bilbao-project-introduction

on two datasets, the number of violinists in both datasets should be approximately the same or comparable. Therefore, we selected performances from 10 of the 22 violinists, i.e. consisting of 10 $\times$ 13 musical scales in total, and then manually annotated the onset position of each note in the selected musical pieces using Sonic Visualiser. Together with the onset annotations, these musical pieces constitute the Selected Solo Sessions (SSC) dataset.

3.3.3 All Solo Clips Dataset

Since we will use the violin dataset to train deep neural networks to identify violinists in Chapter 7, we need as much training data as possible to improve the DNN's performance. Meanwhile, note onset labels are not required for this task. Therefore, we remove the parts of the original recording that do not contain violin performances (e.g., silent parts). The remaining violin performances and the corresponding player IDs are included in the "All Solo Clips" (ASC) dataset.

3.4 Summary

In this chapter, we present the method of violin dataset construction. Three concerto datasets are built from commercial recordings to model the playing style of master players. Considering that there are two scenarios for violinist identification (isolated notes and musical clips), we first segment a certain number of vibrato notes manually from each player's performance to build the IVN dataset, which will be used in Chapter 4. Then, we create a music clip dataset from the concerto recordings and annotate the onset time of each note. This dataset is named SCC and will be used in Chapter 4. Finally, all the concerto clips containing the violinist's performance are used to form the ACC dataset, which will be used to train the deep neural network in Chapter 7.

In addition, to further evaluate our proposed VID methods and provide another scenario for violinist identification, we construct two solo datasets from scale recordings played by professional violinists. The SSC dataset will be used to verify the effectiveness of timbre features on VID in Chapter 5, and ASC will be used to train DNN in Chapter 7.

We summarise the attributes of all datasets in Table 3.3, which includes the music source, name, corresponding violinist identification scenario, size and label for each dataset.

Table 3.3: The Summary of all datasets and their attributes.

Music Source	**Dataset Name**	**VID Scenario**	**Size**	**Label**
Violin Concerto	Isolated Vibrato Notes (IVN)	Individual Notes	248 Notes * 9	Onsets
	Selected Concerto Clips (SCC)	Music Clips	3676 Notes * 9	Onsets
	All Concerto Clips (ACC)	Constant Segments	2 hours * 9	None
Solo Musical Scale	Selected Solo Clips (SSC)	Music Clips	0.5 hour * 10	Onsets
	All Solo Clips (ASC)	Constant Segments	0.5 hour * 22	None

Starting from the next chapter, we will present the proposed violinist identification methods for different scenarios. It begins with the simplest condition, i.e. violinist identification based on isolated notes, in Chapter 4.

Chapter 4

Violinist Identification Using Isolated Notes

4.1 Introduction

As the smallest unit in musical composition is the single note [189], we will attempt to analyse and understand the style of violinists in terms of single notes; note-level features are therefore used for music stylistic modelling. Although it is not common in real life to identify performers based on discrete notes, it could be a starting point for research and has practical applications for specific groups of people. For example, violin students always explore how to express a particular style by imitating the music played by virtuosos. It is easier to start with individual notes rather than an entire piece of music. In addition, it is well known that the variation of melody and rhythm produced by a series of notes can affect the musical style, but whether individual notes can express style remains a question that deserves to be investigated. Moreover, it is crucial for MIR researchers to find relationships between note-level audio features and performer's styles, which will help measure, modify or reproduce musical styles and provide a reasonable basis for further hierarchical

style analysis. To address these issues, this Chapter proposes a method for identifying violinists using isolated notes using hand-crafted audio features and statistical models.

Among the influential factors of music expressions, vibrato plays a vital role in the performance of singing, flute and bowed-string instruments. It is frequently used to enhance selected notes and make them more prominent [62]. As we introduced in Section 2.4.2.2, in violin performance, the property of vibrato is primarily determined by the player's finger movement on the fingerboard, and the rate and extent of vibrato can characterise it. We therefore assume the vibrato extent and rate are good indicators to describe a player's vibrato playing habit and also can be used to identify the player. However, as we reviewed in Section 1.2, previous works attempted violinist identification using pitch, timing or energy features. Such features are generally considered essential for classification, while vibrato features are seldom used for this task. Thus we choose vibrato features as distinguishing factors to identify famous violinists, and the feature extraction method is proposed in Section 4.2.1.2.

Apart from the left-hand playing techniques (such as vibrato), the violinist's style is strongly affected by the bowing gestures such as bow velocity, force, acceleration or bow-bridge distance. In most previous studies [190, 191, 192], the bowing data were acquired and measured by hardware equipment (like sensors). In this case, they can be captured accurately and in real-time, but this usually involves using expensive sensing systems and complex setups that are often intrusive in practice. Furthermore, using a hardware system, it is not feasible to measure bowing parameters directly from audio recordings. However, timbre features extracted from the audio signal are shown to be related to the bowing parameters [39]. Moreover, unlike the raw timbre features that may be influenced by many factors (e.g., instrument or acoustic conditions), timbre variation within each note is more important

for presenting the performer's individual characteristics [193]. Therefore, this Chapter extracts features which reflect note-level timbre variations to model the violinist's playing style and then uses these features to identify violinists.

This Chapter is organised as follows. Methods of feature extraction are presented in Section 4.2, where feature distribution analysis and violinist identification methods are also illustrated. VID experiments are then proposed in Section 4.3, which also includes results and discussions. Finally, we summarise this Chapter in Section 4.4.

4.2 Methods

Figure 4.1 shows the proposed method's overview with four main stages. In the first stage, data pre-processing is firstly applied, followed by audio feature extraction from the IVN dataset. The methods of vibrato feature extraction and timbre feature extraction are separately proposed in Section 4.2.1 as well as 4.2.2. Next, we obtain the distributions of the features to represent the performers' characteristic styles. Such distributions are calculated using three statistical models, respectively, which are shown and analysed in Section 4.2.3. Finally, the similarities among feature distributions are computed to identify performers based on single features and fused features, which will be illustrated in Section 4.2.4.

4.2.1 Vibrato Feature Extraction

To capture vibrato features from polyphonic notes, the first step is obtaining the main melody from the audio signal. We extract the predominant melody from annotated notes using MELODIA [194], so that pitch change within every note can be observed. To avoid noise interference and extract all vibrato features from relevant vibrato data, the melody curve is smoothed before feature extraction. These two steps are denoted *preprocessing* and de-

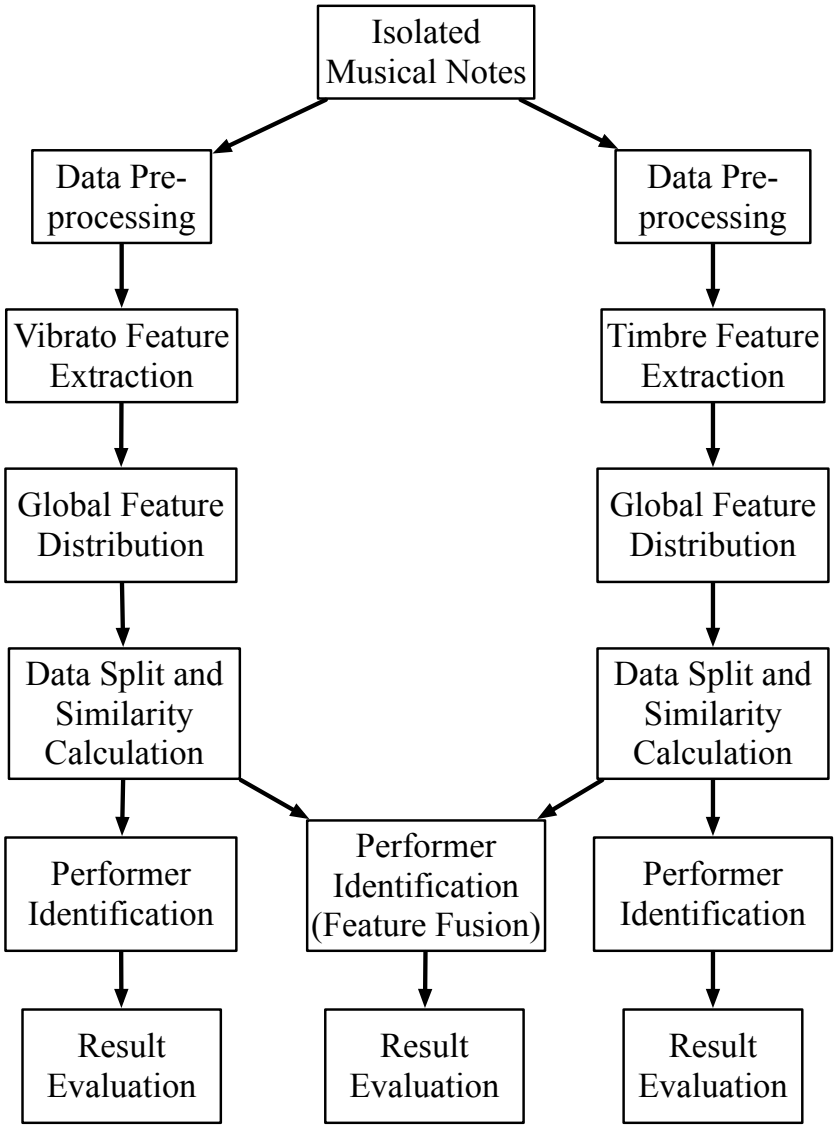


Figure 4.1: Schematic overview of the proposed method for violinist identification using isolated notes.

tailed in Section 4.2.1.1, and the specific vibrato features are introduced in Section 4.2.1.2.

4.2.1.1 Data Pre-processing

MELODIA is an algorithm that outputs the fundamental frequency corresponding to the pitch of the predominant melodic line of a piece of polyphonic music [194]. To capture vibrato features from polyphonic notes, the first step is obtaining the main melody from the audio signal. We extract the predominant melody from annotated notes using MELODIA [194]. The "PredominantMelodia" function implemented in the Essentia library [195] is applied to obtain the predominant melody of polyphonic musical notes. Moreover, MELODIA designates segments without main melody as 0Hz, which are left out from our analysis.

As shown in the middle of Figure 4.2, the curve exhibits noise and artefacts near the peaks and valleys. According to the definition of the rate range and extent range of vibrato in violin playing, it is evident that these high-frequency noises are not generated by vibrato and therefore need to be removed as much as possible before extracting vibrato features. The signal is consequently smoothed to obtain more reliable vibrato features using a zero-phase Butterworth low-pass filter. This avoids the influence of phase delay. The smoothed signal is shown at the bottom of Figure 4.2. But in case of small fluctuations around the boundary, we will address this issue in the following feature extraction process.

4.2.1.2 Feature Extraction

To characterise vibrato, we extract four note-level vibrato features: average vibrato extent (AE), average vibrato rate (AR), standard deviation of vibrato extent (SE), and standard deviation of vibrato rate (SR). All features are computed from the extracted melody.

Vibrato Extent: In every period of the pitch curve, the instantaneous vibrato extent is considered to be the distance of vertical components between an adjacent peak and trough. The average and standard deviation of the

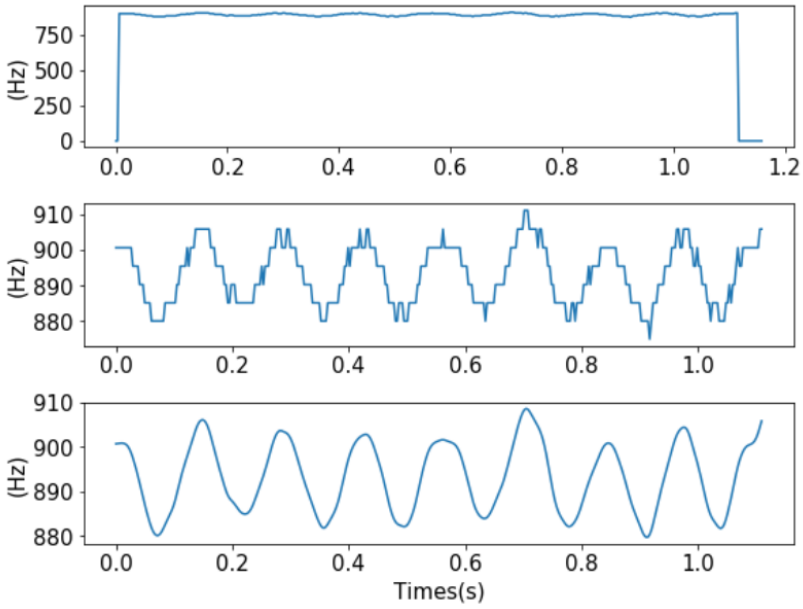


Figure 4.2: Vibrato note pitch curve before and after smoothing.

vibrato extent are calculated from all instant vibrato extent values within a note. First, we find the location of every peak and trough contained in the pitch curve by locating maxima and minima in the smoothed melody data in each period. We then calculate the absolute frequency distance between successive peaks and troughs to obtain the instantaneous vibrato extent. The collection of note-level instant vibrato extents is used to calculate the AE and SE features for all annotated notes.

Vibrato Rate: After obtaining the locations of every peak and trough in the pitch curve of a note, the vibrato rate features can be calculated. We first find the times of peaks and troughs in the pitch curve. The interval between adjacent peaks and troughs is a half period t_h, and the rough instant vibrato rate in every half period is calculated using Equation 4.1. The note-level average vibrato rate (VR) is considered as the mean value of all instant

vibrato rates within a note.

$$VR = 1/(2 * halfperiod). \tag{4.1}$$

Despite the pitch curve being smoothed before feature extraction, oscillations or noise which are not caused by vibrato remains a problem. We consider a heuristic to eliminate the effect of this. In general, the range of the vibrato rate is 2 Hz to 15 Hz, and the range of vibrato extent is between 9 cents and 50 cents, which is also used in [66]. After extracting the rough instant vibrato extent and rate at the note level, we discard values outside these ranges.

4.2.2 Timbre Feature Extraction

In this section, we present the method of timbre feature extraction. The data was pre-processed before feature extractions to reduce timbre differences caused by different recording conditions. It includes silence removal and loudness normalisation, which will be introduced in Section 4.2.2.1. The timbre feature selection and extraction are then presented in Section 4.2.2.2.

However, in addition to the individual interpretation of performers, the timbre features are influenced by many other factors (e.g. instrument, recording conditions), and the fundamental frequency of a note impacts the values of some features (e.g. spectral centroid). In this case, raw timbre features are not necessarily comparable between performers. We therefore assume the performer's characteristic playing primarily produces the variation of timbre features within a note. The method of calculating feature variation is presented in Section 4.2.2.3.

4.2.2.1 Data Pre-processing

To make the extracted features more comparable among performers, we remove silent regions in each music clip, and then apply loudness normalisation based on the EBU standard [196]. All steps were completed in Audacity[1].

4.2.2.2 Feature Extraction

We select features that are either commonly used in the literature in related tasks, or have been validated in the context of violin bowing technique recognition in [197].

Six timbre-related features are considered. One feature represents spectral moments (Spectral Centroid), three features describe the shape of the spectrum (Mel-Frequency Cepstral Coefficients, Spectral Bandwidth [198], Spectral Contrast [94]), and two temporal features (RMS energy and Zero-crossing rate). Details of these features were introduced in Section 2.4.4, further details and discussion can be found in related papers [199, 200, 201, 202, 203].

The segmented notes in IVN dataset are first divided into short overlapping frames (f_s=44.1 kHz, frame length = 2048, hop size = 512), and all features are extracted at frame level and summarised at note level. For each note, the note-level MFCCs and spectral contrast are multi-dimensional vectors, whereas other features are single-dimensional.

4.2.2.3 Feature Standardisation

We calculate the *z-score* of each feature vector at the note level, which aims to standardising features by removing the mean and scaling to unit variance. The standard score z of each sample x in the feature vector is calculated using Equation 4.2:

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

$$z = \frac{x - u}{s}, \tag{4.2}$$

where u is the mean value of the feature vector, and s is the standard deviation. For single-dimensional features like Spectral Centroid, RMS, ZCR and Spectral Bandwidth, the feature vector can be standardised directly using this formula. But for multi-dimensional features like MFCCs and Spectral Contrast, the feature vector is standardised at the dimension level, and the original dimensionality remains unchanged before and after the standardisation.

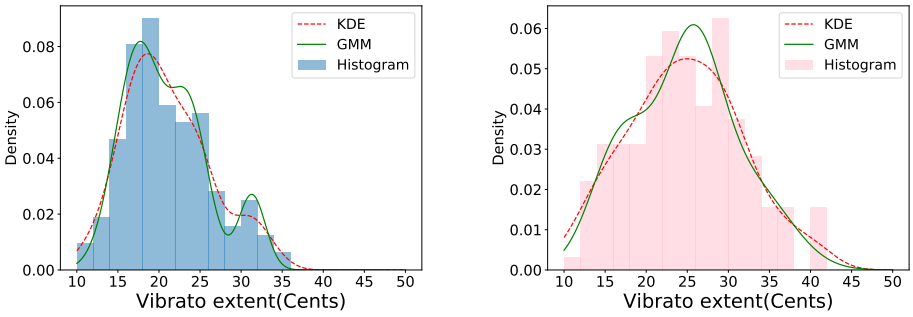
For clarity, a summary of the features used in this Thesis and their abbreviations are listed in Table 4.1.

Table 4.1: Summary of features and abbreviations

Original Feature Name	**Shortened Name**
Average Vibrato Extent	AE
Average Vibrato Rate	AR
Standard Deviation Vibrato Extent	SE
Standard Deviation Vibrato Rate	SR
Combination of all Vibrato features	VC
Spectral Centroid	SC
Spectral Contrast	SCT
Zero Crossing Rate	ZCR
Spectral Bandwidth	SB
Root Mean Square Energy	RMS
Mel-Frequency Cepstral Coefficients	MFCCs

4.2.3 Feature Distribution Estimation

When different performers play the same music piece, they typically express vibrato or timbre styles in their respective performances. Therefore, we model the characteristics of each performer using the distribution of the extracted features. Three statistical models, including Histogram, Kernel Density Estimation (KDE) and Gaussian Mixture Model (GMM), are separately



(a) Heifetz (b) Mutter

Figure 4.3: Distribution of two performer's average vibrato extent

used to model these distributions, assuming that these provide compact representations of the violinists' style, which we can use later for identification. Section 4.2.3.1 and Section 4.2.3.2 show distributions based on vibrato features and timbre features, from which we can observe the characteristic of performers' feature distributions.

4.2.3.1 Vibrato Feature Distributions

Figure 4.3 shows how the global distribution of average vibrato extent for Heifetz and Mutter differs, for example. We can easily see that the highest density of the vibrato extent distribution appears between 15 cents and 20 cents for Heifetz, but it is 20 cents to 25 cents and 29 to 30 cents in Mutter's performances. In addition, Heifetz's performances have no vibrato greater than 35 cents, whereas the maximum vibrato extent reaches above 40 cents in Mutter's. This shows that Heifetz prefers to use the vibrato on a smaller scale, but Mutter's vibrato extents are broader. Based on similar observations for several performers, we can assume that the feature reflects an essential aspect of the vibrato characteristics of every performer.

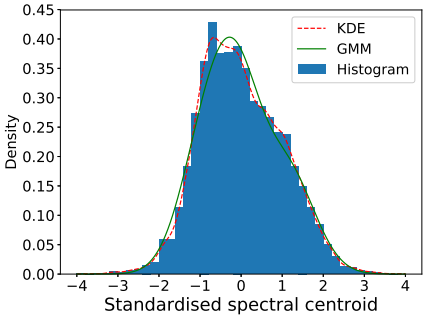
In Figure 4.3, the red line shows the Gaussian kernel to estimate the kernel density of average vibrato extent data from Heifetz and Mutter as

well, and the curve of the two distributions shows similar properties to histograms. We also train a 3-component Gaussian Mixture Model to estimate the distribution of the data. Their PDF curves are shown in Figure 4.3 using continuous (green) lines. The number of components in these models is selected using empirical observation, i.e., the distributions do not generally exceed three modes, so the GMM represent the histograms and kernel densities well. Given these curves, we can observe the continuous distributions of features for each performer, and their differences should reflect individual characteristics.

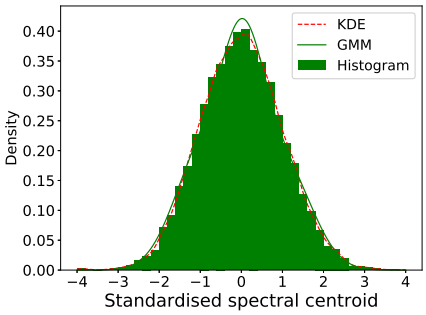
4.2.3.2 Timbre Feature Distributions

Followed by the observations above, Figure 4.4 compares the three timbre feature distributions of Heifetz and Mutter using the three statistical models mentioned above. Figure 4.4(a) and Figure 4.4(b) show the global distributions of "Standardised Spectral Centroid", it is easy to notice that the general shape of the two distributions is different, where the peak of the Heifetz's distribution appears on the left side, and it has a gentler slope to its right side. In contrast, the entire distribution of Mutter looks closer to a symmetrical pattern, with the highest peak occurring near the point of origin in the horizontal direction. A similar phenomenon can be found in Figures 4.4(c) and 4.4(d), which represent the distribution of the ZCR features of the two players. The bottom two figures compare the distributions of the 3rd coefficient of MFCCs (which is denoted as MFCCs(c3)) for two performers. Although their discrepancies are not as easy to discover as the previous two sets of plots, it is still possible to observe the differences between the two distributions in terms of slope, width, etc.

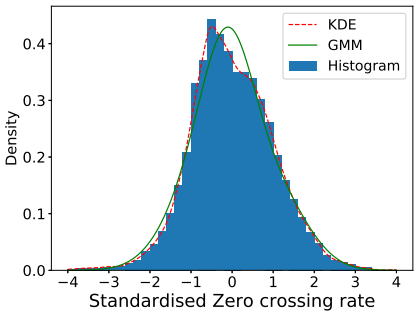
Similar to what we mentioned in Section 4.2.3.1, the global feature distributions are assumed to represent the performer's playing style, and differences among the distributions can be used for identifying performers.



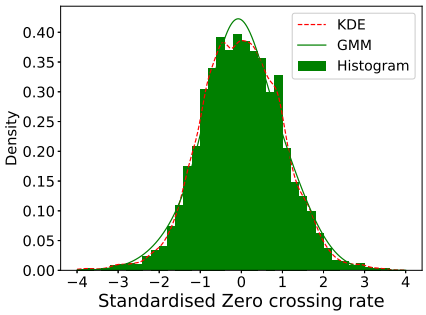
(a) Heifetz



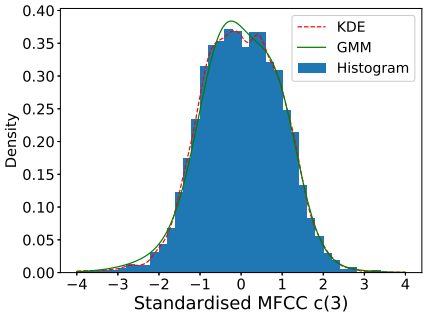
(b) Mutter



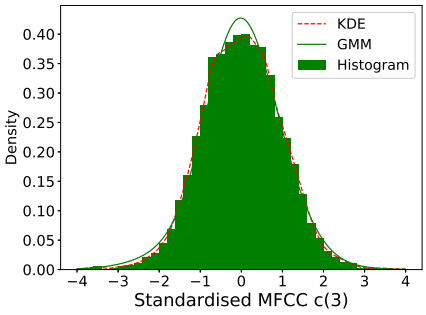
(c) Heifetz



(d) Mutter



(e) Heifetz



(f) Mutter

Figure 4.4: Distribution of two performers' timbre features.

4.2.4 Violinist Identification using Feature Distributions

4.2.4.1 Violinist Identification based on Single Feature

To quantify the differences among feature distributions, we calculate the similarity of distributions of each given feature for all performers using the Kullback-Leibler (KL) divergence [129] shown in Equation 4.3. This corresponds to the likelihood ratio between two distributions and tells us how well the probability distribution Q approximates the probability distribution P by computing the cross entropy minus the entropy.

$$D_{KL}(P \,||\, Q)) = \sum_{i} P(i) log(\frac{P(i)}{Q(i)}) \tag{4.3}$$

For classification, the KL divergence can be calculated between each vibrato feature distribution or timbre feature distribution of an unknown performer and every known performer in the dataset. The smaller KL divergence, the greater similarity. Therefore finding the minimum divergence between an unknown and known performer should help to identify the unknown performer.

4.2.4.2 Violinist Identification based on Feature Fusion

Due to the low number of features, we sidestep the use of complex feature selection methods. We use the linear combination with equal weights to fuse similarity estimates for the distributions of different features summarised in Table 4.1. During the evaluation, leave one group out cross-validation (LOGOCV) with 15 folds (movement level) or five folds (concerto level) is used to calculate the KL divergence between the training set and test set for every group of data. The similarity estimates of feature distributions in every fold are combined for the different kinds of features using the approach shown in

Equation 4.4:

$$D_{KL_{overall}} = \sum_{n=1}^{|\Theta|} w_n D_{KL_{\Theta_n}} \tag{4.4}$$

where $\Theta = \{V_1, V_2, V_3, V_4, T_1, T_2, T_3, T_4, T_5, T_6\}$ with $V_1, ..., T_6$ denoting the sets of feature distributions corresponding to four kinds of vibrato features (AE,AR,SE,SR) and six kinds of timbre features (SC, MFCCs, ZCR, SCT, SB, RMS) computed separately. D_{KL_Θ} means the normalised KL divergence values in each cross-validation process. All corresponding weights w_n are set to one in the current implementation. Moreover, how the features are fused is not unique; we can combine any number of features to compute the overall KL divergence. Next, we verify the applicability of this method to the identification of violinists and test the accuracy for different performers. The design of the experiment and the results are discussed in Section 4.3.

4.3 Experiments

We assess the proposed identification method using LOGOCV and show the classification results using macro F-score and confusion matrices for all performers. In this section, we first introduce the experiment setup and data preparation. We then present the results of using different features to identify violinists in Section 4.3.2. Finally, we discuss the result in Section 4.3.3.

4.3.1 Experimental Setup

To avoid overlapping music pieces between the training and test sets, we separately use movement-level and concerto-level LOGOCV in the classification experiment. In each fold, we designate recordings of one movement or one concerto played by all nine performers as the test set, while the remaining recordings are placed into the training set. This eliminates piece overlap

between the training and test sets (Each concerto in the test set includes three movements). Single labels are assigned in the test set at also two levels: concerto-level and movement-level. We then compute the KL divergence between each feature's distribution from the test performer and the same for every performer in the training set. Similarity results based on distributions of each feature are obtained between the test performer and every performer in the training set.

4.3.2 Results

4.3.2.1 Baseline Methods

To assess our proposed feature and violinist identification method, we set two groups of baseline methods. First, since KNN and SVM are frequently used as classifiers in performer identification tasks [23, 144], we use these two models as baseline methods based on the IVN dataset; the dataset is randomly split into training set and test set at a ratio of 80% and 20%. In the training phase, feature vectors of four vibrato features along with the corresponding player ID in the training set are considered as input to train the machine learning models. In the test phase, the feature vectors extracted from the test set are used to evaluate the trained models, and the F-measure results are shown in Table 4.2. Second, a vibrato feature extraction method based on "wavelet scattering" is proposed, and the SVM is also applied to identify violinists using this feature based on the IVN dataset. Due to the scope of this Thesis, we simply place the corresponding experimental results in Table 4.2 without presenting further details, but they can be checked in [204].

It is observed that among these baseline methods, wavelet scattering gives the best results, while KNN performs the worst. These results suggest that the original vibrato feature vector and the direct use of machine learning

Table 4.2: Violinist identification result using baseline models on vibrato features.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
KNN(n=3)	0.173	0.157	0.133	0.187	0.162	0.160
SVM	0.227	0.212	0.186	0.238	0.227	0.229
Wavelet Scattering	0.38	0.24	0.23	0.43	0.29	0.27

models cannot classify violinists well. In the following sections, we will present detailed results of the proposed methods and compare them with these baseline methods.

4.3.2.2 Results on Vibrato Features

As for our proposed method, we first designate a performer as test performer. Then separate each performer's data in movement level, so that for each test player, we can get 15 distributions for each feature. Furthermore, since there are four vibrato features, 60 distributions for one test performer can be obtained to present vibrato characteristics. Similar mechanism can be applied when the data are split at concerto level, where five distributions for each feature can be obtained.

In this experiment, we first select the test performer, and then designate all annotated notes from one movement (or one concerto) played by this performer as the test data. The same movement (or concerto) that other performers play is left out, whereas the remaining pieces from all performers (including the test performer) are placed in the training set. We then compute the KL divergence between each feature's distribution from test data and the same features for every performer in the training data. The similarity results for vibrato characteristics based on four features can be separately obtained between the test performer and every performer in the training set. The smaller the KL divergence, the greater the similarity, there-

fore we treat the performer that corresponds to the minimum value as the identified performer with each feature.

Finally, we obtain the result based on fused vibrato features, which is computed by Equation 4.4 with $\Theta = \{V_1, V_2, V_3, V_4\}$. These are denoted as "combination vibrato features" (VC).

Table 4.3: Violinist identification results based on vibrato features using different statistical models (Movement-level).

F1-score / Feature / Model	VC	AE	AR	SE	SR
Histogram	0.312	0.123	**0.216**	0.117	0.152
KDE	**0.338**	**0.140**	0.175	0.155	0.156
GMM	0.298	0.123	0.206	**0.173**	**0.168**

Table 4.4: Violinist identification results based on vibrato features using different statistical models (Concerto-level).

F1-score / Feature / Model	VC	AE	AR	SE	SR
Histogram	0.313	0.129	**0.215**	0.073	0.194
KDE	**0.426**	**0.262**	0.162	0.100	**0.210**
GMM	0.364	0.144	0.135	**0.137**	0.150

Table 4.3 shows the violinist identification result in F1-score using three statistical models separately when the data are split in movement level. At the same time, Table 4.4 compares the results when the data are split at the concerto level. The abbreviations of the Table can be checked in Table 4.1. These results are better than the baselines shown in Table 4.2. Furthermore, both results indicate that VC performs better than any single feature, while KDE produces the best performance. Therefore, Table 4.5 shows the detailed results obtained using KDE with the two data split methods.

Table 4.5: Violinist identification results using vibrato feature KDE and two data split methods.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
VC	**0.373**	**0.333**	**0.339**	**0.538**	**0.400**	**0.426**
AE	0.162	0.170	0.140	0.277	0.289	0.262
AR	0.205	0.200	0.175	0.164	0.178	0.162
SE	0.156	0.170	0.155	0.113	0.111	0.100
SR	0.170	0.163	0.156	0.230	0.200	0.210

4.3.2.3 Results on Timbre Features

Next, we also assess the same method using timbre features on the IVN dataset. Similarly, to avoid overlapping musical segments between the training and test, the data are split at movement and concerto levels, respectively. We then compute the KL divergence between each feature's distribution from test data of test performer and training data of all performers, and LOGOCV is also applied. The performer who obtained the minimum KL divergence in the training set is considered as the target performer.

To validate the performance of different statistical models, we evaluate the violinist identification methods using Histogram, KDE and GMM separately with two data split strategies, shown in Table 4.6 and Table 4.7. The histogram outperformed the other two distributions on most features, regardless of which data split method was used. Moreover, only when using MFCCs, the KDE outperform other models, but the advantage was not significant, especially when the data was segmented at concerto level.

We therefore present detailed results in Table 4.8 to illustrate the performance of violinist identification based on each timbre feature, when using histograms to calculate the distribution. Among these features, SC and MFCCs performed best in identifying violinists. SCT and SC also show promising results in identifying violinists (higher than 0.5 in F1-score), which confirms these features help identify violinists. In addition, the results based on SB

Table 4.6: Violinist identification results based on timbre features using different statistical models (Movement-level)

F1-score / Feature / Model	SC	SCT	SB	MFCCs	RMS	ZCR
Histogram	**0.608**	**0.539**	**0.470**	0.547	**0.466**	**0.191**
KDE	0.372	0.306	0.351	**0.628**	0.286	0.136
GMM	0.182	0.153	0.122	0.151	0.120	0.081

Table 4.7: Violinist identification results based on timbre features using different statistical models (Concerto-level)

F1-score / Feature / Model	SC	SCT	SB	MFCCs	RMS	ZCR
Histogram	**0.823**	**0.695**	**0.646**	0.864	**0.651**	**0.285**
KDE	0.523	0.520	0.503	**0.869**	0.519	0.214
GMM	0.214	0.172	0.138	0.255	0.134	0.097

and RMS are also higher than the random baseline, suggesting that these features can also help to distinguish performers. However, the ZCR performs worst, which indicates it may not reflect the player's playing style as good as other timbre features.

Table 4.8: Violinist identification results using timbre feature distributions (histogram) and two data split methods.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
SB	0.478	0.474	0.470	0.685	0.644	0.646
SC	**0.615**	**0.615**	**0.608**	0.855	0.822	0.823
SCT	0.556	0.536	0.539	0.732	0.688	0.695
RMS	0.494	0.459	0.466	0.706	0.688	0.651
MFCCs	0.561	0.548	0.548	**0.881**	**0.867**	**0.865**
ZCR	0.211	0.194	0.191	0.243	0.230	0.232

Finally, we attempt to identify violinists by fusing 3 or 4 of the best-performing timbre features using Equation 4.4. Four groups of features were chosen, and their results are shown in Table 4.8. The first group consists of

three features $\{SC, SCT, MFCCs\}$, denoted as $TC3$ (*Timbre Combination* 3); the second group consists of four features $\{SB, SC, SCT, MFCCs\}$, denoted as $TC4$ (*Timbre Combination* 4). In addition, $TC5$ and $TC6$ represent the results of using a fusion of the best performing 5 and 6 timbre features to identify violinists. The results based on these fused timbre features are shown in Table 4.9. It is obvious that the performance of the proposed VID system becomes more powerful as the number of fused features increases, and this phenomenon is also independent of the data split strategy.

Table 4.9: Violinist identification results using fused timbre feature distributions (histogram) and with data split methods.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
TC3	0.761	0.756	0.752	0.908	0.867	0.869
TC4	0.775	0.763	0.760	0.950	0.933	0.931
TC5	0.773	0.763	0.764	0.950	0.933	0.931
TC6	**0.794**	**0.778**	**0.778**	**0.950**	**0.933**	**0.934**

4.3.2.4 Results on Fused Features

After presenting results based on a single category of features, we fuse different categories of features to identify violinists. As we mentioned in Section 4.2.4.2, the feature fusion method is not unique, so we can select any feature from the extracted feature set. However, the performance of every single feature should be considered to obtain better results. According to the results above, VC performs better than any single vibrato feature, while MFCCs and SC are more discriminative of the violinist's playing style than ZCR. Therefore, we will further evaluate the results for different combinations of fused features.

To maintain the generalisation of the method, we prefer to use only one statistical model to obtain the distribution of all features. In this experiment, the histogram is chosen due to its best performance on most timbre features,

and its performance on vibrato features is also better than GMM. In addition, the feature fusion is also computed using Equation 4.4.

Table 4.10: Violinist identification results using timbre feature histogram and two data split methods.

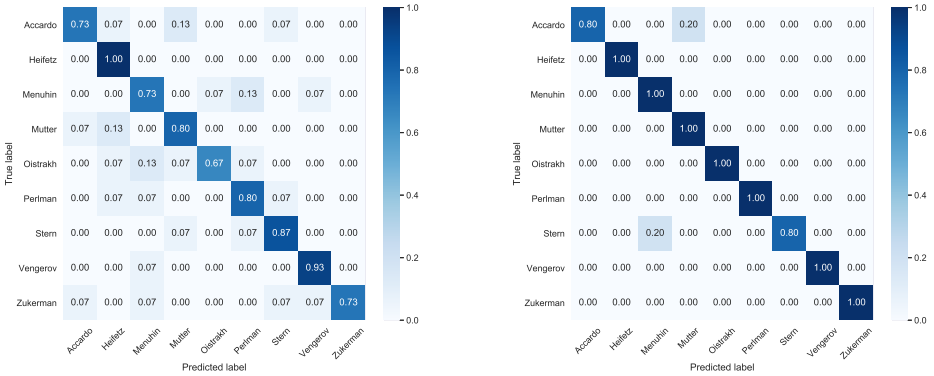
	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
$3FF$	0.772	0.763	0.762	0.900	0.889	0.887
$4FF$	0.806	0.800	0.798	0.918	0.889	0.889
$5FF$	0.796	0.792	0.792	0.937	0.933	0.933
$6FF$	0.778	0.770	0.771	0.963	0.956	0.955
$7FF$	**0.820**	**0.807**	**0.806**	**0.963**	**0.956**	**0.956**

We first combine VC with SCT and MFCCs together, whose result is shown as "3 Feature Fusion ($3FF$)" in Table 4.10. Then, we fuse four features consisting of VC, MFCCs, SCT and SC, denoted as $4FF$; VC combined with MFCCs, SCT, SC, and RMS is then named $5FF$. Finally, the VC fused with MFCCs, SC, SCT, RMS, and SB are shown as $6FF$ in Table 4.10. The $7FF$ means feature fusion with all extracted features together. We can observe that regardless of the data split strategy applied, $7FF$ performs best in terms of F1-score, and the violinist identification results become better as the number of fused features increases.

To visualise the results intuitively, we present confusion matrices for violinist identification using $7FF$, where Figure 4.5(a) shows the results based on movement-level data split, while Figure 4.5(b) shows the results from concerto-level data split.

4.3.3 Discussion

We will discuss the above results in three aspects: the selection of features, the data split strategy and the choice of statistical models. From the results of Section 4.3.2.2, we find that the individual vibrato features do not show good discrimination between violinists. But when the four vibrato features are



(a) Data Split in Movement-level (b) Data Split in Concerto-level

Figure 4.5: Violinists classification using $7FF$.

fused, the results are much better. However, according to the results shown in Section 4.3.2.3, we notice that all timbre features except ZCR show promising results, especially Spectral Centroid and MFCCs. This suggests that the designed timbre features are more helpful in describing the violinist's style than the vibrato features, and the fused timbre features perform better than any single one. Moreover, we obtained a higher F1-score by fusing different kinds of features, which further demonstrates that using more features helps improve the algorithm's performance.

Based on these observations, we notice that the single vibrato feature is less discriminative for violinists, while fused vibrato features perform better. One possible reason is that characterising vibrato is a complex process, and using only one single feature cannot represent a player's vibrato style. Importantly, however, although the results are not very satisfactory, vibrato can be used to identify violinists, suggesting that virtuoso performers play vibrato with their individual style, and the style can be measured. Furthermore, the timbre characteristics are reasonable for identifying violinists, from which we can learn that the timbre variation within notes is a good indicator for representing the violinist's style.

In terms of the data split strategy, all data are split at the movement level and concerto level separately, resulting in different folds for cross-validation and a different amount of data for each fold. Not surprisingly, the results are better when the data are split at the concerto level. One reason is that when using 5-fold cross-validation, more data in the test set is used to calculate the feature distribution, which makes it more representative of the performer's style and also reduces the chance of over-fitting. However, when using 15-fold cross-validation, some movements only contain four annotated notes (see Table 3.1). The distributions obtained based on these movements were not sufficiently reflective of the player's characteristics, and the corresponding results are therefore less favourable.

We applied three statistical models, including histogram, KDE and GMM, to model the distribution of features. As shown in Figure 4.3 and Figure 4.4, the histogram and KDE provide a clearer picture of the differences between performers, and they yield better results for the classification of violinists. However, the GMM does not perform well when modelling the timbre feature distributions. In Figure 4.4, the differences between players are not clearly visible. This may be related to the parameters we set to fit the GMM (e.g. the number of components), but due to the scope of this Thesis, this part will be placed in future work. Finally, with more features being fused, time cost of the algorithm is increased. Among the three distribution models, histogram is the most time efficient, while the other two algorithms are more time-consuming due to the complicated procedures of data fitting (as we show in Section 2.5.1) and KL divergence calculation.

4.4 Summary

This Chapter investigates the influence of vibrato and timbre on violinist recognition using the IVN dataset. We first design and extract four vibrato

features, including AE, AR, SE and SR, to describe the performer's vibrato characteristics and then use three statistical models to obtain the distribution of each feature to represent the individual playing style of the performer. Next, the data are split at movement and concerto levels, respectively, and the similarity between the training and test data distributions is calculated to identify the violinists. Finally, a feature fusion method using these four vibrato features is proposed, and the results show that the fused features work better than any single vibrato feature.

Similarly, six timbre features are extracted to present the performer's timbre characteristic, and the distributions of these features are applied to model and identify violinists. The results show that most of the timbre features work better than vibrato features, and an F1-score of 0.865 is obtained based on MFCCs, and 0.934 can be found based on TC6. Finally, feature fusion methods are proposed, using two feature categories to identify violinists. The best performance is obtained from $FF7$, which further suggests that our proposed features and models are beneficial for identifying violinists, and the results become better as the number of fused features increases.

Although timbre features perform very well in identifying violinists, some uncertain questions still need to be answered. For example, since timbre is intuitively influenced by recording conditions and instruments, can our approach identify violinists based on the individual style of the performer rather than on other factors? To address this problem, in the next Chapter, we will verify the effectiveness of designed timbre features for describing the violinist's playing style.

Chapter 5

The Effectiveness of Timbre Features for Identifying Violinists

5.1 Introduction

In Chapter 4, we extract six note-level timbre features to model performers' timbre characteristics. Although timbre features perform well in identifying violinists based on isolated notes, it is unclear whether the designed features reflect the stylistic characteristics of the performer, or the acoustic characteristics resulting from other factors. For example, professional or well-known violinists generally have their preferred instruments, and early research suggests that timbre features can be used to distinguish different violins [205]. To explore whether our timbre feature-based algorithm identifies the performer's style or the instrument's characteristics, we will design and conduct some experiments based on the SSC dataset in this Chapter. The performance in this dataset is recorded by ten violinists on 13 violins, and all performers were invited to complete the recording in the same studio, so

that the influence of recording conditions could first be ignored. Next, two experiments will be conducted, including violinist and violin identification, to verify whether timbre features are primarily influenced by the performer's style or the instrument.

We first introduce the timbre feature extraction method in Section 5.2. Next, Section 5.3 illustrates the experiments and results. Finally, the chapter is summed in Section 5.4.

5.2 Methods

The proposed method in this chapter is outlined in Figure 5.1. To avoid the influence of recording conditions, the data are pre-processed before the timbre feature extraction, which will be illustrated in Section 5.2.1. Next, we obtained the distribution of each feature for each performer, and some examples of distribution plots are shown in Section 5.2.2 to facilitate our observation of the differences among performers' characteristic playing. Finally, we conduct two experiments, including violinist identification and violin identification, to investigate the effectiveness of the designed timbre features for identifying violinists, which will be presented in Section 5.2.3 and Section 5.2.4.

5.2.1 Feature Extraction

Although all performances in the SSC dataset were recorded in the same studio and with the same microphone, it is difficult to guarantee that the distance and orientation between each performer and the microphone will remain constant. Therefore, to make the extracted features comparable among performers, we remove silent regions in each music clip, and then normalise loudness based on the EBU standard [196]. All steps were completed in Audacity [1].

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

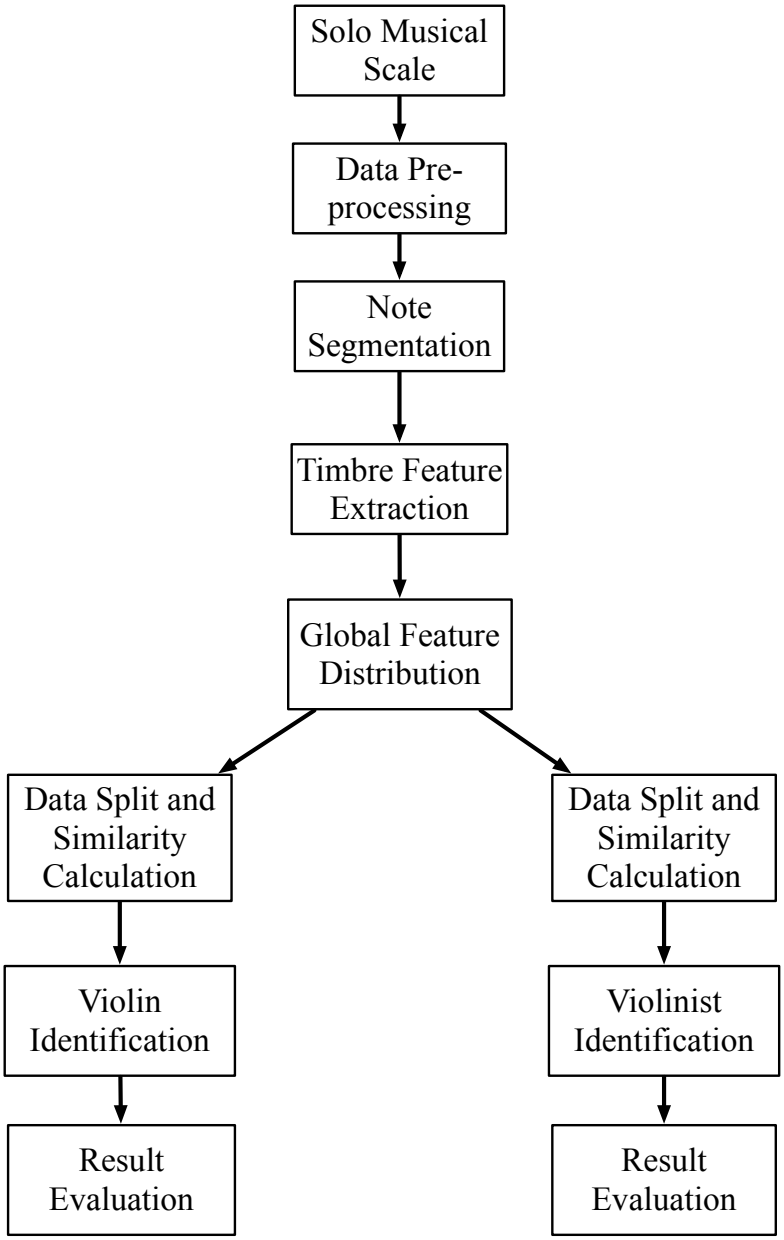


Figure 5.1: The outline of timbre feature validation method based on SSC dataset.

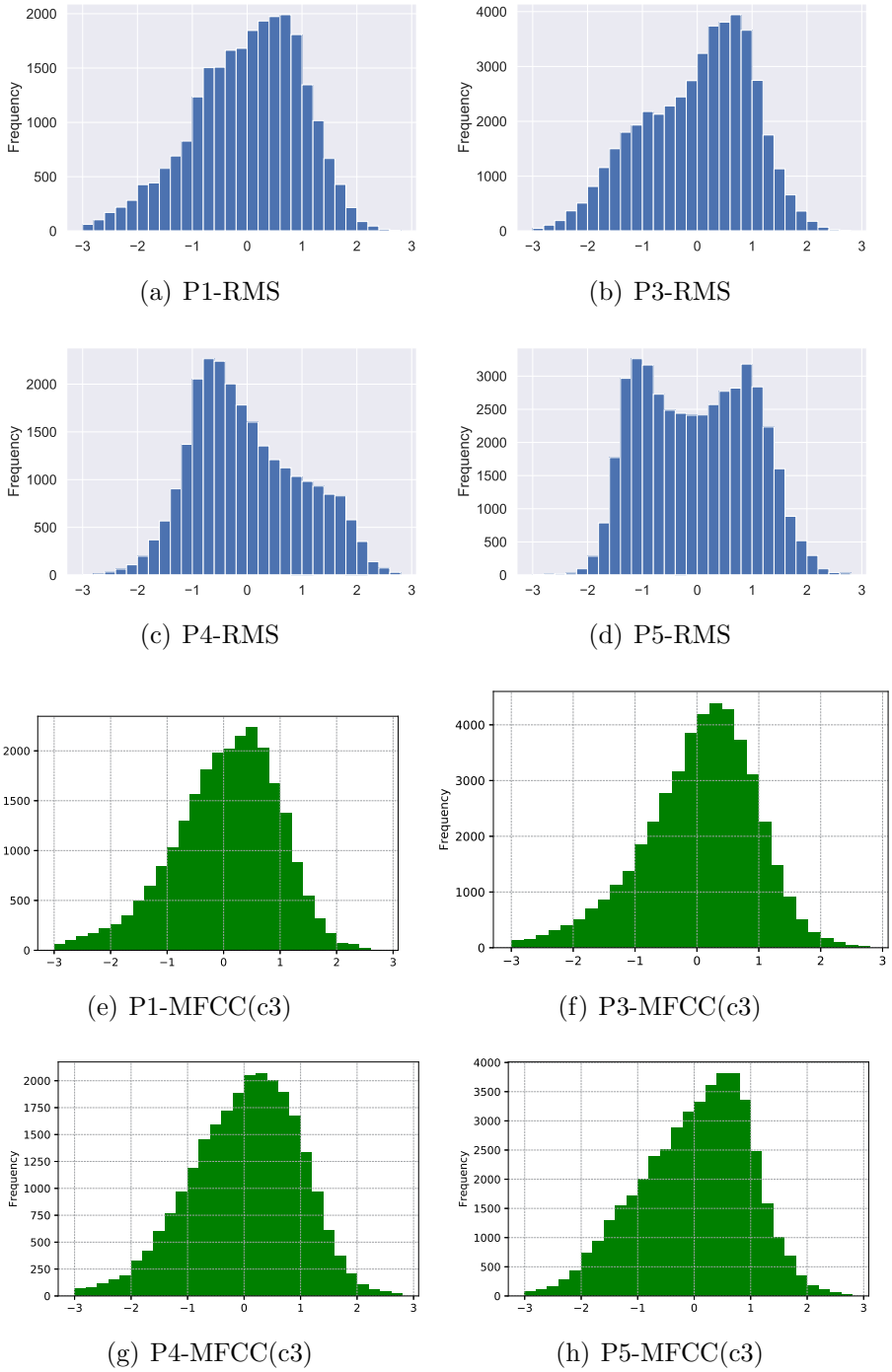
To make the number of violinists comparable with that in the concerto dataset, we select music recordings played by ten violinists and then annotate the onset times of each note from the solo dataset (See Section 3.3). The segmented notes are firstly divided into short overlapping frames (f_s=44.1 kHz, frame length = 2048, hop size = 512). The timbre features are extracted from each frame and grouped at the note level. For each frame, we extract six timbre features, including Spectral Bandwidth (SB), Spectral Centroid (SC), Spectral Contrast (SCT), RMS Energy (RMS), Mel-Spectral Cepstral Coefficients (MFCCs), and Zero-crossing Rate (ZCR). The calculation method of each feature is described in Section 2.4.4.

However, we care about the timbre variation within a note rather than raw timbre features. Hence the feature standardisation is also applied at the note level, which is the same as presented in Section 4.2.2.3.

5.2.2 Feature Distribution

Due to the histogram performing well in modelling violinist's timbre characteristics in Chapter 4, we also use the histogram to calculate timbre feature distributions. For multi-dimensional features, we use multi histograms to model such data distributions at the dimension level. For example, since the MFCC is 12-dimensional, there are 12 histogram distributions to present one performer's style based on the MFCC feature.

Figure 5.2 shows the global distribution of the RMS and 3^rd MFCC coefficient (MFCC(c3)) features for four performers in the solo dataset. At the same time, the x-axis means the range of standardised features, and the y-axis presents the frequency. We abbreviate "Perfomer1" as "P1", and the same abbreviation is applied to all ten performers. The shapes of such distributions are different, as seen in the figure. For example, the histogram of standardised RMS features for "P1" is less sharp than that of P3. The skew and the mass centre of the distributions also differ, which can be seen



(a) P1-RMS

(b) P3-RMS

(c) P4-RMS

(d) P5-RMS

(e) P1-MFCC(c3)

(f) P3-MFCC(c3)

(g) P4-MFCC(c3)

(h) P5-MFCC(c3)

Figure 5.2: Distribution of four performers' standardised RMS and MFCC(c3) feature in the solo dataset.

in the differences between "P4" and "P1" and "P3". The number of modes may also be different, which can be seen with "P5", for example, where there are two peaks in the histogram. This indicates the performer might prefer to play each note using more flexible dynamics. The histogram in Figure 5.2(e)-5.2(h) present the MFCC(c3) distributions from the same four performers. The sharpness, position of the highest bar, and slope differ among such distributions. Based on similar observations across different performers and features, we assume that such features indeed reflect an important aspect of the performer's timbre characteristics.

5.2.3 Violinist Identification using Timbre Features

To quantify these differences, we calculate the similarity of distributions of each feature for all performers using the Kullback-Leibler (KL) divergence [129], presented as $D_{KL}(P||Q)$. This corresponds to the likelihood ratio between two distributions and tells us how well the probability distribution Q approximates the probability distribution P.

$$D_{KL}(P \,||\, Q) = \sum_{i} P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{5.1}$$

For classification, the KL divergence is calculated between each timbre feature distribution of an unknown performer and every known performer in the dataset. Minimum divergence identifies the unknown performer. Classification experiments using this approach are presented in Section 5.3.1.

5.2.4 Violin Identification using Timbre Features

Next, we classify violins based on proposed feature distributions to investigate the influence of music instruments on timbre features. Since there are 13 violins and each violin is played by all ten performers, we divide the whole dataset into 13 groups. Each group contains ten music pieces played

on each violin. Then the KL divergence between each timbre feature distributions from an unknown violin and every known violin in the dataset. The minimum divergence is also applied to identify the unknown violin, and the detailed experiment process is proposed in Section 5.3.2.

5.3 Experiments

In this section, we first investigate how the method performs for identifying violinists using music scales. Next, we conduct violin identification using a similar approach to verify whether violins influence the distribution of timbre features. For each experiment, we assess the proposed method using *leave one group out cross-validation*. Macro F-score is used as an evaluation metric for all performers or violins in the dataset.

5.3.1 Violinist Identification

In the SSC dataset, a musical scale (which contains approximate 37 notes) by each performer plays on each violin, and there are 130 (10 violinists $\times$ 13 violins) musical scale recordings in total. In the experiment, we firstly select a random performer as the test performer, then designate one musical scale that played with a random violin from such performer as test data. Other musical pieces played with this test violin from other performers are left out, and the remaining pieces from all performers (including the test performer) are placed in the training set.

Then, we compute the KL divergence between each feature's distribution from the test performer and the same features for every performer in the training data. The similarity results for timbre characteristics based on six features can be obtained between the test performer and every performer in the training set. The smaller the KL divergence, the greater the similarity, therefore we treat the performer that corresponds to the minimum value as

the identified performer with each feature.

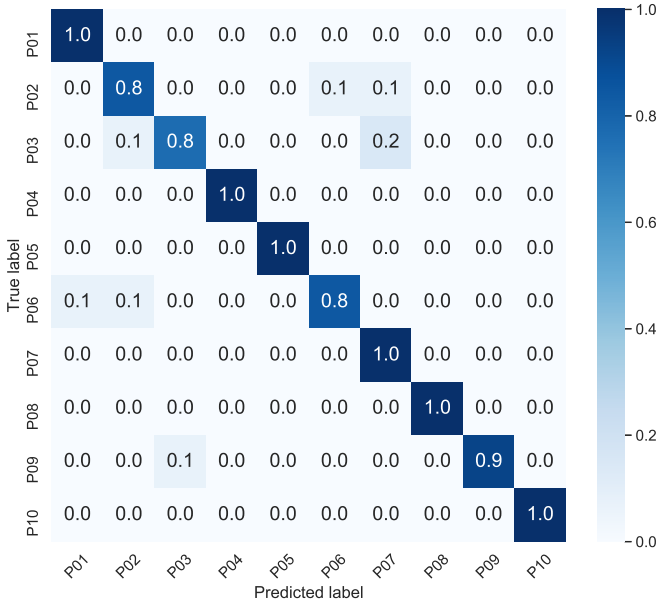


Figure 5.3: Normalised confusion matrix for violinist identification using standardised MFCC feature distributions.

Based on the leave one group out cross-validation (LOGOCV), we get the similarity of timbre features between every two performers in the dataset and the performer identification result using each feature. Table 5.1 shows the macro F-score result of violinist identification using each feature distribution separately. MFCC work best among all features, which suggests the feature has good discrimination power on performers. The confusion matrix is shown in Figure 5.3 corroborating our observation. It can be seen in Table 5.1 that the classification result based on Spectral Contrast is the second best, which shows 0.908 in F1-score. RMS performs best among time domain features, whereas the zero-crossing rate is less helpful for identifying violinists.

Table 5.1: Violinist identification results based on each timbre feature using solo dataset.

Feature	Precision	Recall	F1-score
Spectral Centroid	0.459	0.438	0.439
RMS	0.789	0.777	0.781
Spectral Bandwidth	0.370	0.369	0.365
Zero-crossing rate	0.243	0.246	0.235
Spectral Contrast	0.918	0.908	0.908
MFCC	**0.941**	**0.938**	**0.937**

5.3.2 Violin Identification

Similarly, to identify violins, we first select a test violin and then designate one musical scale played by a random performer on this violin as test data, other music recordings played by the same performer are all left out. Next, the remaining music pieces are put into the training set and split into 13 groups according to the violin index of the music recordings.

Each type of standardised timbre feature is used separately to build a histogram, and the KL divergence between the test data and training data from each violin is then calculated. Since there are 13 violins in total, the KL divergence is calculated 13 times, and the violin corresponding to the smallest value is considered as the result of recognition. In addition, 10-fold leave one group out cross-validation is applied because there are ten performers in each group of training data. Finally, the violin identification results are obtained and shown in Table 5.2.

From the violin identification results, we find that the F1-scores are around or under 0.1, close to the random baseline. Figure 5.4 shows the confusion matrix of the violin classification based on MFCC feature distributions, which presents that no classes of violins can be identified correctly by using the proposed features and approach. These results further verify that our designed feature and classification approach can characterise performers' styles rather than the properties of the violins.

Table 5.2: Violin identification results based on each timbre feature using solo dataset.

Feature	Precision	Recall	F1-score
Spectral Centroid	0.096	0.108	0.097
RMS	0.109	0.092	0.056
Spectral Bandwidth	0.043	0.135	0.062
Zero-crossing rate	**0.117**	**0.115**	**0.114**
Spectral Contrast	0.076	0.085	0.076
MFCC	0.123	0.108	0.083

Although it is suggested that timbre is related to the violin's structure and sound quality [206], the timbre features proposed in this Thesis are primarily determined by the violinist rather than the violin. It is mainly because the features demonstrate the note-level variation of timbre rather than the original timbre characteristics in the musical performance. Generally, the timbre variation within a note is primarily produced by the player, and it is not strongly related to the instrument's characteristics, which allows us to use this characteristic to model the player's style.

5.4 Summary

In this chapter, two experiments are designed to explore whether the proposed timbre features and the VID method (presented in the previous chapter) identify violinists based on their individual playing styles or the instrument's characteristics. The results show that our proposed features and methods perform well in identifying performers, but they do not help distinguish instruments. However, the more common performer identification scenario is based on short musical clips rather than isolated notes. How to identify violinists in this case? In the next Chapter, we will propose a new feature to identify violinists from short musical clips, while validated timbre features are also used to model performer characteristics.

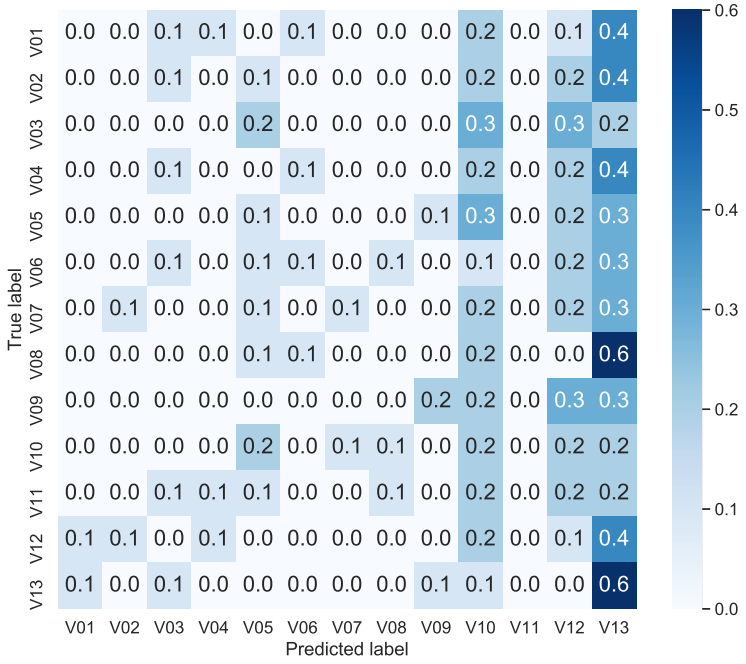


Figure 5.4: Normalised confusion matrix for violin identification using standardised MFCC feature distributions.

Chapter 6

Violinist Identification Using Short Music Clips

6.1 Introduction

In Chapter 4, four vibrato features and six timbre features are extracted to model the characteristic playing style of violinists. The selected timbre features are then validated in Chapter 5, and the results show that such features can reasonably and objectively model performers' individual playing, regardless of the instrument they play. However, for most listeners, a more common and practical scenario for identifying performers is using musical segments rather than isolated notes. Therefore, this Chapter will present a method for identifying violinists based on short music clips using the SCC dataset.

Although timing characteristics among different performers' performances have been investigated in previous work [24], it is unclear how useful these features are in quantitatively describing and identifying a performer's style. Therefore, in this chapter, two timing features are devised to describe the temporal preferences of violinists, and then are used to identify them. The

feature extraction method and the corresponding VID experiments are shown in Section 6.2.1 and Section 6.2.2 respectively.

Furthermore, as the effectiveness of designed timbre features in identifying violinists has been verified in previous chapters, we will extract the same features and evaluate their performance based on the SCC dataset. Finally, we will fuse these two types of features, and the violinist identification results and discussion will be presented in Section 6.3.3.

6.2 Methods

Although each piece of music has its own rhythmic and emotional characteristic, performers can vary the note duration and tempo according to their preferences. For example, some players slow down the tempo for a fast-paced piece to make each note intelligible, while others may increase the speed to enhance the emotional expression. It is commonly believed that "note" is the smallest unit in a music piece, but the duration of a single note is random and has little meaning in describing a player's timing preference. We assume that when performers play a given music piece and the onset time of the first note is set as "0s", other notes' onsets would be different. These differences are accumulated and superimposed as the music progresses, which provides a good presentation of performers' global expressive timing characteristics. Therefore we design and extract two note-level features: Onset Time Deviation (OTD) and Note Duration (ND), then analyse the timing feature distributions across music pieces to model the performer's timing characteristics.

Figure 6.1 shows the outline of this chapter. We first present methods of timing feature extraction, including feature calculation and normalisation. Next, the feature distributions are shown and analysed in Section 6.2.2. In addition, six validated timbre features are extracted separately from the SCC

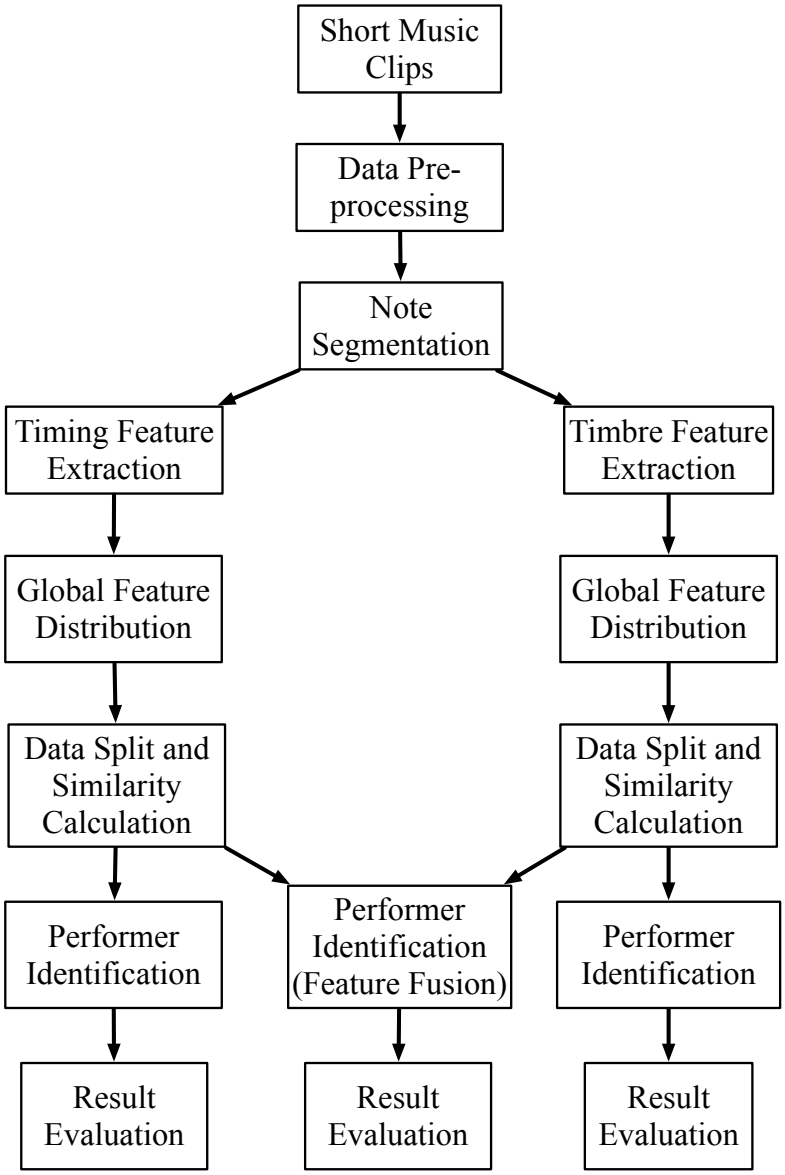


Figure 6.1: Schematic outline of the proposed method for violinist identification using music clips

dataset, and the distributions of these features are shown and illustrated in Section 6.2.2.3. Finally, these two types of features are fused to distinguish violinists, which will be presented in Section 6.2.3.

6.2.1 Feature Extraction

6.2.1.1 Onset time deviation

For a given music piece, to calculate the deviation of note onset times among different performers, the reference onset time of each note should be first obtained. There are two possible approaches to getting reference: score-based note onset times, or mean note onset times across all performances. The former method provides a standard reference that is not influenced by any existing performances. For the latter, we can assume that averaging removes most of the performers' expressive timing and individual interpretation, except for a generally accepted interpretation of the piece, where such interpretation exists. This method also avoids the need to align the audio with the score. We apply both methods in this Thesis to investigate which method better captures the performer's characteristics.

We first use the average note onset time as the reference time to illustrate the feature extraction method. For each selected music clip, as Figure 6.2 shows, the first note is aligned in time, which is all set as $0s$. The alignment is then applied to other notes, and the mean onset time of each note from all violinists' performances can be calculated as the reference time. This is followed by calculating onset time deviations from this reference for each performer to characterise expressive timing. For example, a score is shown at the top of Figure 6.2. The different vertical bars indicate note duration from different performers. The vertical dashed lines are the average onset times of each note in this piece, which are regarded as the "reference onset time". Then the time distance between each actual onset time and the reference time

is the onset time deviation. However, the onsets are conditional to previous notes and to the general phrases for a single performer. We therefore apply a "0-1" normalisation to all onset time deviations for each performer using the Equation 6.1, where $Onset_i$ is the raw onset deviation of the i^{th} note, $OnsetNorm_i$ is the normalised onset deviation, max and min denote the maximum and minimum values of all onset deviations of the performer.

$$OnsetNorm_i = \frac{onset_i - min}{max - min} \tag{6.1}$$

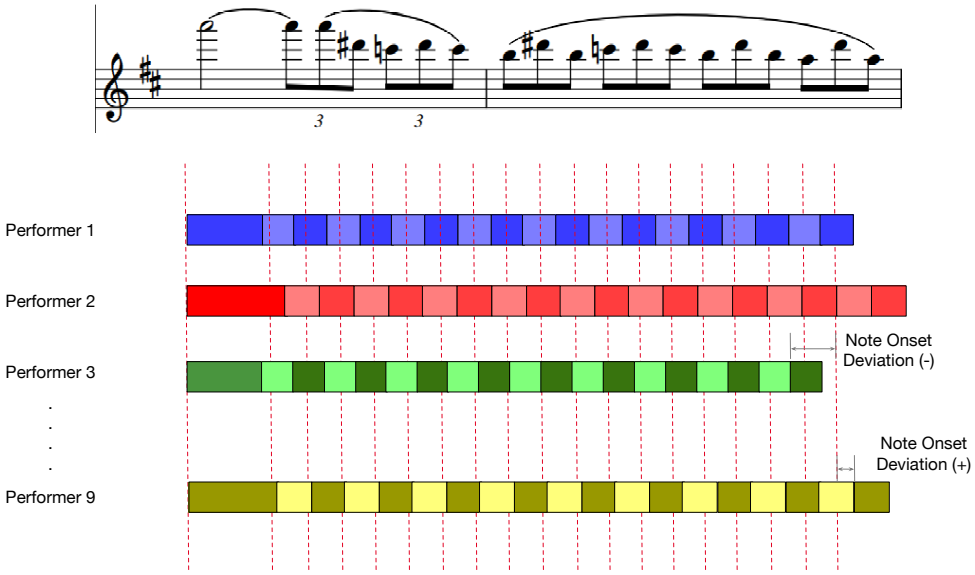


Figure 6.2: Expressive timing feature extraction

A similar mechanism is used to calculate the OTD feature based on the reference time obtained from the music score. To find the note onsets, we first download audio files of concerto synthesised from the score[1] and then split the music into short segments identical to those in the SCC dataset. Next, we manually annotated the onset time of each note, and these annotations are used as the reference times. Finally, each performer's OTD feature can be calculated using the method presented above. To distinguish two OTD

[1]https://musescore.com/

features calculated base on different reference times, the one using "average time" as reference is abbreviated as "OTD_AVG", and the other is named "OTD_Score".

6.2.1.2 Note Duration

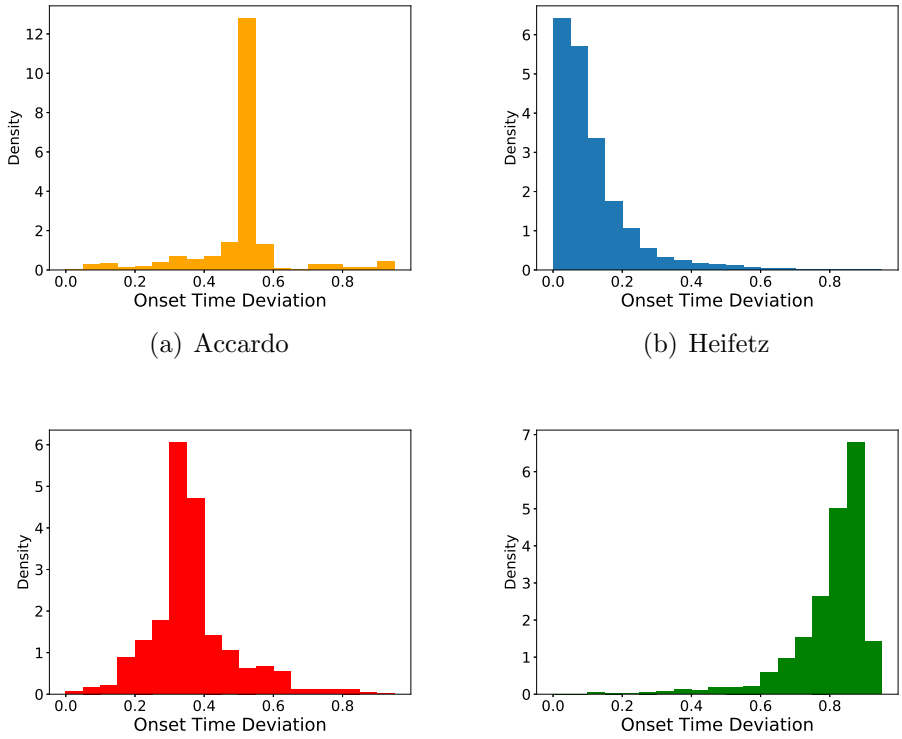
Apart from the onset deviation, the note duration (ND) is also considered as a feature to describe the performer's style. Although the note duration highly depends on the music pieces as well as note types, the same note can be performed in different duration by different performers. This feature is computed by Equation 6.2.1.2, where T_i means the duration of the i^{th} note, and the $Onset_{i+1}$ and $Onset_i$ denote the time stamp of the onset times of the i^{th} note and the $(i+1)^{th}$ note, respectively.

$$T_i = Onset_{i+1} - Onset_i$$

6.2.2 Feature Distribution

Following the idea of previous chapters, we assume that the global distribution of audio features across musical pieces can characterise certain aspects of a performer's playing style. This section presents the characteristic distributions of OTD and ND, and analyses the temporal characteristics of the performers observed from these distributions. In addition, the distribution of timbre features is also shown and discussed, which will be illustrated in Section 6.2.2.3.

As the histogram performed better in previous chapters for feature statistical modelling, in this Chapter, all distributions are obtained using histograms, which also allows us to focus more on discussing and observing the performance of each feature for violinist identification.



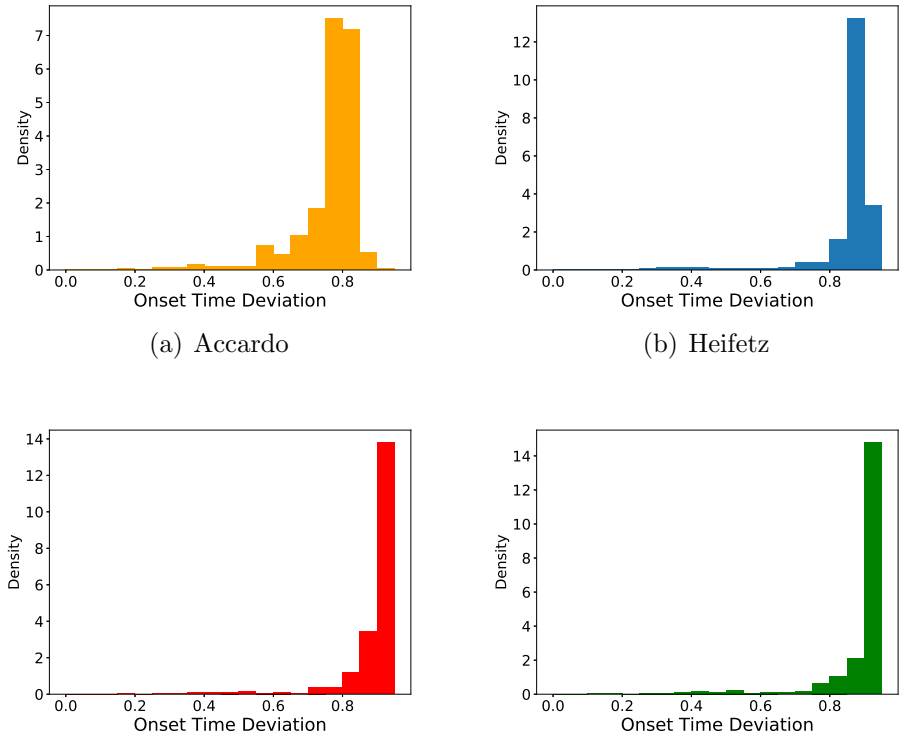
(a) Accardo (b) Heifetz

(c) Menuhin (d) Mutter

Figure 6.3: Distribution of four performers' OTD_AVG features.

6.2.2.1 OTD Distribution

Figure 6.3 shows the OTD_AVG feature distributions of four violinists using the histogram. There is a skewed right histogram for feature distribution from Heifetz's performance, showing that the actual note onset times in Heifetz's performance are mostly earlier than the reference time, indicating he prefers to play the music much faster than the average speed. However, opposite phenomena can be observed in Mutter's performance, which illustrates that Mutter usually plays the music at a slower tempo. Although the Bell-shaped histograms can be observed from distributions of Accardo and Menuhin, the skew and centre of their highest bar are different. This shows that the two performers might prefer to play each note at dynamic but different speeds.



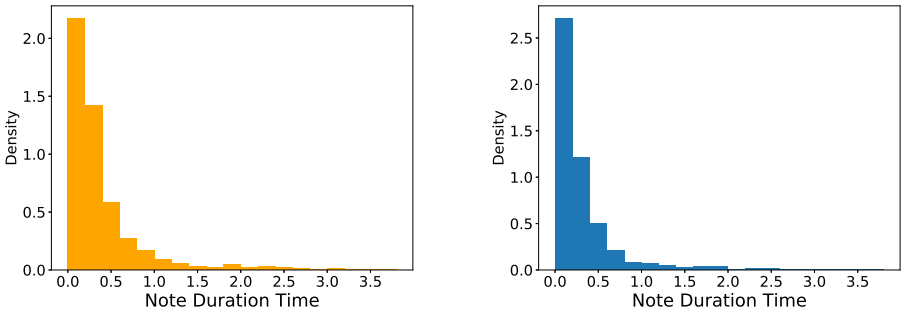
(a) Accardo (b) Heifetz

(c) Menuhin (d) Mutter

Figure 6.4: Distribution of four performer's OTD_Score features.

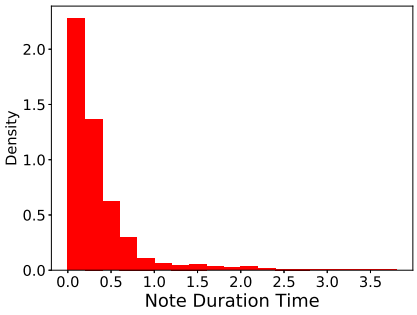
Based on similar observations from distributions from other performers, it is clear that the distribution of OTD_AVG features shows excellent discrimination among performers.

Next, we show the distribution of Onset_Score from the mentioned four performers. As can be seen in Figure 6.4, most bars frequently occur on the right side, which indicates the actual note onset times are always slower than the reference time of the score. The highest bar is observed on 0.8 at the x-axis from Accardo's histogram, whereas the peaks of the other three histograms are seen at 0.9 or 1.0. However, the differences between the distribution from Menuhin and Mutter are not as significant as observed in Figure 6.3.

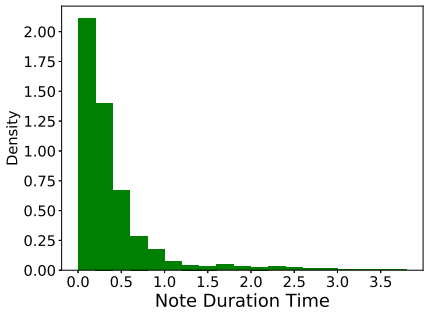


(a) Accardo

(b) Heifetz



(c) Menuhin



(d) Mutter

Figure 6.5: Distribution of four performers' ND features.

6.2.2.2 ND Distribution

The distributions of ND features are presented in Figure 6.5, where the x-axis means the duration time and the y-axis denotes the density. It can be seen that all of the histograms are skewed to the right, which makes us cannot detect the differences among performers' timing preferences based on these distributions. Nevertheless, we will examine the performance of this feature in identifying violinists in Section 6.3.2.1.

6.2.2.3 Timbre Feature Distribution

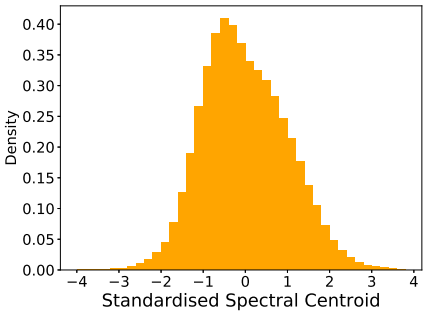
In addition to timing features, we also extracted timbre features that effectively identified violinists in the previous Chapters. Six note-level fea-

tures including Spectral Centroid (SC), Spectral Bandwidth (SB), Spectral Contrast (SCT), Root Mean Squared Energy (RMS), Mel-frequency cepstral coefficients (MFCCs) and Zero-crossing Rate (ZCR) are extracted in frame level, and then standardised in note level. The feature calculation methods are introduced in Section 2.4.4, and the feature standardisation method is presented in Section 4.2.2

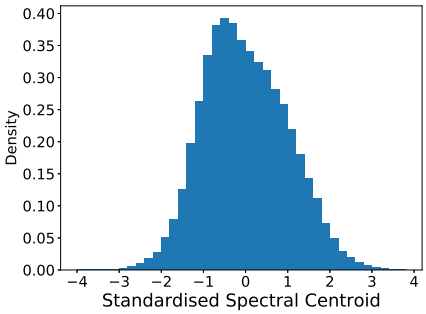
Figure 6.6 shows the distribution of two timbre features separately. Figure 6.6(a)-6.6(d) exhibit the global distributions of the standardised spectral centroid based on the performance data from four players, with different colours used to indicate different players for better differentiation. The general shape of these distributions is very similar, but it can be found that Accardo's distribution shape is the most kurtic, while Heifetz's is less sharp, and the kurtosis for Menuhin's is the smallest. In addition, the shape of Mutter's distribution, especially the right-hand slope, is slightly different from the other three, which indicates her playing characteristics. On the other hand, the histogram in Figure 6.6(e)-6.6(h) present the distributions of 3^{rd} coefficient of MFCCs (denote as MFCCs (c3)) from four performers, the sharpness, position of the highest bar, and slope are slightly different among such distributions. Based on similar observations across different performers and features, we assume that such features can also reflect the performer's individual timbre characteristics based on the SCC dataset.

6.2.3 Violinist Identification

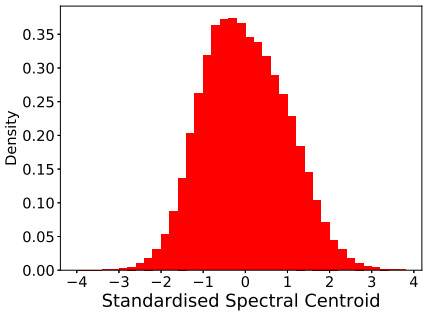
In order to quantify these differences, we calculate the similarity of distributions of each feature for all performers using the Kullback-Leibler (KL) divergence [129], presented as $D_{KL}(P||Q)$. This corresponds to the likelihood ratio between two distributions and tells us how well the probability distribution Q approximates the probability distribution P.



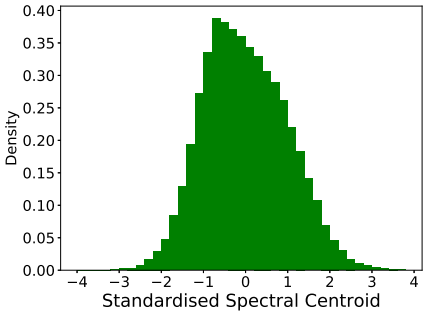
(a) Accardo_SC



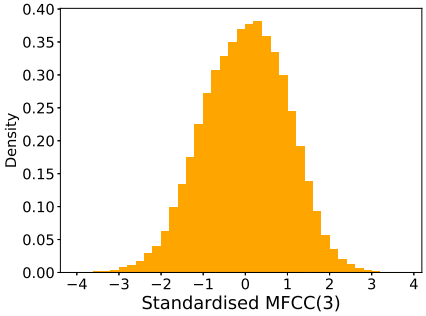
(b) Heifetz_SC



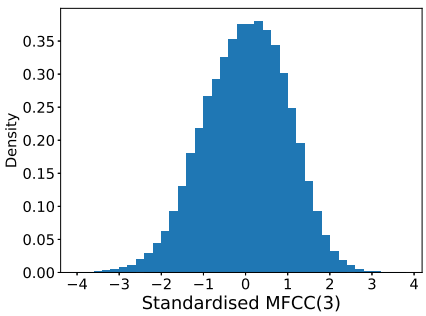
(c) Menuhin_SC



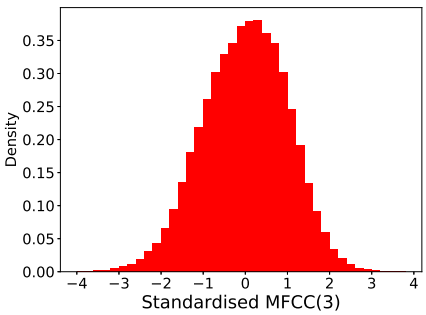
(d) Mutter_SC



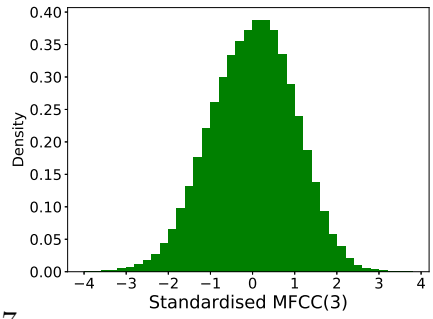
(e) Accardo_MFCCs (c3)



(f) Heifetz_MFCCs (c3)



(g) Menuhin(c3)



(h) Mutter_MFCCs (c3)

Figure 6.6: Distribution of four performers' Timbre features.

$$D_{KL}(P \,||\, Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{6.2}$$

For classification, the KL divergence is calculated between each feature distribution of an unknown performer and every known performer in the dataset, and the minimum divergence identifies the unknown performer. Details of the VID method are described in Section 4.2.4, and experiments using this method on the SCC dataset are described in Section 6.3.

6.3 Experiments

In this section, we first apply the violinist identification method using our designed timing features, investigating how they perform for identifying players using the SCC dataset. Next, to verify whether the timbre feature distributions help classify violinists using the same dataset, we use the timbre feature distribution to model and identify violinists, which will be presented in Section 6.3.2.2. Finally, the two categories of features are fused into different groups, and the corresponding results are shown and discussed in Section 6.3.2.3.

In this section, we first identify violinists using the designed timing features to understand how they perform on the SCC dataset. Next, to verify whether the timbre feature distributions also contribute to the violinist identification on this dataset, the results of applying the timbre features are also presented in Section 6.3.2.2. Finally, we fuse these two types of features in different ways, and the corresponding results for VID are presented and discussed in Section 6.3.2.3.

We assess the proposed method for each experiment using leave one group out cross-validation (LOGOCV) and evaluate the classification results using the macro F-score metric.

Table 6.1: Violinist identification results using timing feature distributions with two data split methods.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
OTD_AVG	**0.739**	**0.726**	**0.730**	**0.836**	**0.800**	**0.797**
OTD_Score	0.353	0.385	0.356	0.367	0.363	0.365
ND	0.119	0.148	0.124	0.117	0.178	0.100

6.3.1 Experimental Setup

In this experiment, the dataset is split in movement and concerto levels, the same as we mentioned in Section 4.3.1. Therefore, different data strategies are applied to assess the violinist identification approach, where 15-fold LOGOCV is used when data is split at movement level, and 5-fold LOGOCV is used when data is split at concerto level.

6.3.2 Results

6.3.2.1 Violinist Identification on Timing Feature Distributions

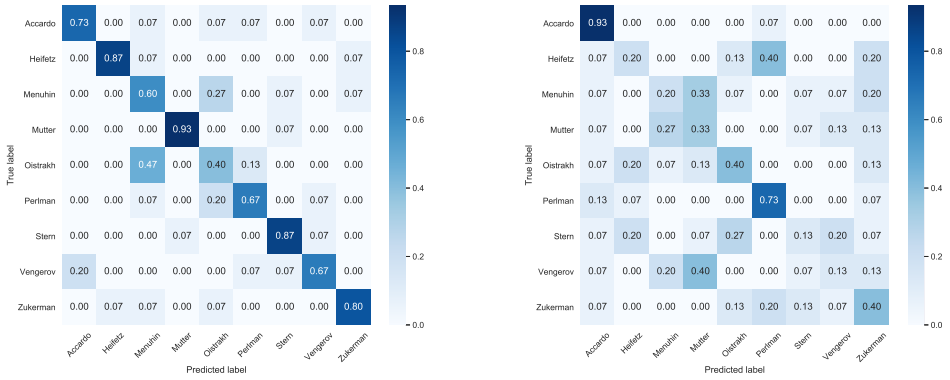
Table 6.1 shows the results of violinist identification using three timing features. It is noticed that OTD_AVG performs the best, achieving F1-scores of 0.730 and 0.797 when the data is segmented in movement level and concerto level, respectively. In addition, OTD_Score does not perform as good as OTD_AVG, but it also has a certain ability to discriminate performers, obtaining F1-scores of around 0.36 with either data split method. However, ND performs the worst, yielding an F1 score of only around 0.1, close to the random baseline. It suggests that this feature is practically unable to identify violinists in this case.

To further investigate the performance of OTD_AVG and OTD_Score, we present the confusion matrix of violinist identification based on those two features in Figure 6.7. Although Table 6.1 shows a reasonable result

of OTD_Score, the confusion matrix based on this feature indicates that its recognition for different players is highly variable, i.e. it is not a feature that is consistent enough to be effective for VID. For example, the discrimination of OTD_Score is good for Accardo and Perlman, but very poor (even with random baselines) for other players. However, the confusion matrix based on the OTD_AVG distribution gives an entirely different result, which presents good discrimination for most players. Based on these observations, we find that the selection of reference note onset time is essential when calculating the OTD, which determines the generalisation of the feature. As shown in Figure 6.4, the OTD_Score distributions for all three players except Accardo is generally a left-skewed histogram, which indicates that their actual note onset times are often much slower than the reference, referring not a good differentiator among the performances of these performers. When the average note onset time is used as the reference time, the players' differences are magnified, making it easier to identify violinists. Nevertheless, it is easy to discover that Menuhin's performance is confused with Oistrakh's, which means they have similar timing feature distributions. Therefore, we can conclude that this approach based on timing features is promising, as it works well for identifying most performers in our dataset, but it may still produce some confusion between certain performers.

6.3.2.2 Violinist Identification based on Timbre Feature Distributions

In addition to timing features, we also evaluate the performance of violinist identification based on timbre features. Table 6.2 shows the F-measure results based on six timbre features separately, where MFCCs perform best no matter whether the data is split in movement-level or concerto-level, with corresponding F1-scores of 0.326 and 0.351. The SCT is the second best, where F1-scores of 0.258 and 0.274 are observed. However, the worst results



(a) OTD_AVG (b) OTD_Score

Figure 6.7: The Confusion Matrices of Violinist Identification based on OTD_AVG Distribution and OTD_Score Distribution.

are obtained based on SB, with F1-scores of 0.140 and 0.133, showing the feature distributions do not help characterise violinists. The F1-scores of other features such as SC, RMS and ZCR are mostly clustered around 0.2, suggesting that they can classify violinists' playing to some extent, but their effects are limited.

Table 6.2: Violinist identification results using timbre feature distributions (histogram) and two data split methods.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
SB	0.179	0.194	0.170	0.130	0.178	0.140
SC	0.235	0.236	0.235	0.226	0.215	0.214
RMS	0.170	0.167	0.165	0.207	0.193	0.192
ZCR	0.226	0.207	0.198	0.137	0.135	0.136
SCT	0.324	0.283	0.302	0.306	0.282	0.274
MFCCs	**0.341**	**0.333**	**0.326**	**0.352**	**0.363**	**0.351**

6.3.2.3 Violinist Identification based on Fused Feature Distributions

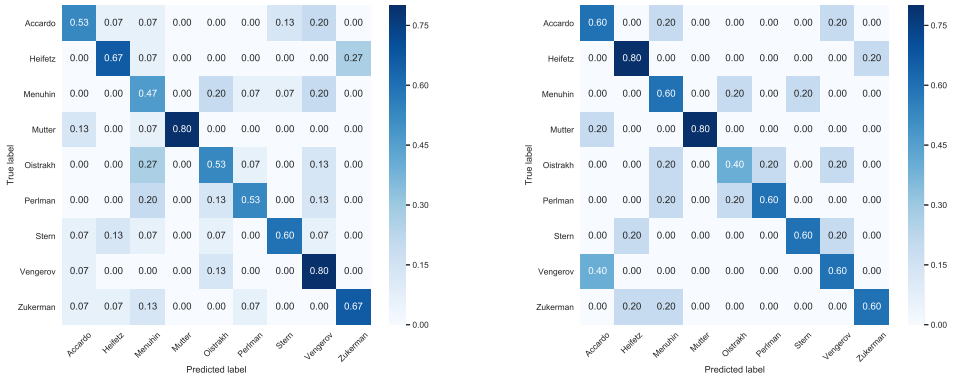
Finally, we conduct a violinist identification experiment using fused features. The feature fusion method was introduced in Section 4.2.4.2, which is calculated using the Equation 4.4. After evaluating the performance of every feature in the above sections, we select the best timing feature OTD_AVG and the best three timbre features, including MFCCs, SCT, and SC, into the feature fusion group, and the results are shown in Table 6.3. We firstly combine the OTD_AVG feature with MFCCs and SCT, and the result is shown as "Feature Fusion 3 (FF3)". Then, we fuse the OTD_AVG with MFCCs, SCT and SC, whose results are presented FF4 in Table 6.3. Finally, all timbre features and OTD_AVG are fused to identify violinist, whose results are illustrated as "FF7" in Table 6.3 as well.

Table 6.3: Violinist identification results using fused feature distributions (histogram) and two data split methods.

	Movement-level			Concerto-level		
Feature	Precision	Recall	F1-score	Precision	Recall	F1-score
FF3	**0.692**	**0.659**	**0.668**	**0.701**	**0.644**	**0.642**
FF4	0.654	0.622	0.630	0.648	0.600	0.597
FF7	0.639	0.615	0.618	0.604	0.622	0.590

It can be seen that the F1-score and precision of FF3 are the best, and the corresponding confusion matrix is shown in Figure 6.8. According to the Figure, each performer can be correctly classified. Although the overall F-measure results are a bit lower than using OTD_AVG, the discrimination for every performer is higher than using this timing feature only (e.g., Menuhin and Oistrakh are confused with each other in Figure 6.8(a), but in Figure 6.8 the issue is mitigated). So there is a noticeable improvement in discrimination using the feature fusion. However, the results of FF7 are much worse than FF3, suggesting that it is not the case that the more features that are

fused, the better the results for VID. The main reason is that the designed timbre features may be influenced by factors such as accompaniment based on the SCC dataset, which does not accurately reflect the player's style and introduce noise into the violinist identification results. In addition, as more features are fused, the computational cost becomes more expensive.



(a) Movement-level (b) Concerto-level

Figure 6.8: The Confusion Matrices of Violinist Identification based on FF3 Distribution using two data split strategies.

6.3.3 Discussions

According to the results presented by the above experiments, we can discuss them in two aspects. Firstly, with respect to the feature selection, we find that OTD_AVG is the best performing timing feature for violinist identification, while OTD_Score is the second best, and ND is the worst. However, based on the observation of the confusion matrix in Figure 6.8(b), we found that when using OTD_Score to identify violinists, the identification results for different performers varied considerably, which also indicates the feature has poor generalisation on VID. Furthermore, although the OTD_AVG has better discrimination for each performer, there is still some confusion between specific performers. On the other hand, according to the results in Table 6.2, we

find that the timbre features perform much worse than OTD_AVG. Among these timbre features, MFCCs and SCT are the best, while SB is hardly helpful for identifying violinists. This is inconsistent with the results we obtained in the previous two chapters, which may be due to the influence of the accompaniment on the extraction of timbre features, making the features describe not only the characteristics of the violinist's performance, but also the characteristics of the timbre variation in the accompaniment. Finally, it is not the case that the more features are fused, the better results can be obtained. The designed timbre features cannot describe the violinist's playing characteristics well (although it still helps) and do not serve to increase the amount of distinguishing information of performers when these features are fused.

Secondly, in Chapter 4, we found better results for concerto-level-based data splits than for movement-level-based ones, but this conclusion is no longer applicable in this chapter. Compared with the IVN dataset, the SCC dataset contains a substantially larger amount of data, allowing reliable feature distributions to be obtained in each fold of the cross-validated test set, regardless of which strategy is used to segment the data. It makes the results obtained by the movement-based data split method better. However, as the same performer collaborates with different orchestras when they perform different concertos (See the Appendix A), it is likely to make the timbre differences among concertos greater. Moreover, as each concerto comes from a different composer who wrote them in a different style and era, it makes the music of each concerto very different. Furthermore, since our dataset contains only five concertos and each test set contains only music clips from one concerto, it is not a good indication of performers' global characteristics. Therefore, the concerto-based data split method yields worse VID results.

6.4 Summary

This Chapter proposes a method for identifying violinists from short concerto films. Two expressive timing features are designed and extracted, and the global distribution of such features across all performances is obtained to present the performer's style. Moreover, the timbre features validated in previous chapters are also applied to model performers' timbre characteristics based on the SCC dataset. The two categories of audio features are separately used and then fused to classify violinists. The results show that OTD features perform better, while ND feature performs poorly. In addition, the timbre features are helpful to recognise violinists from the SCC dataset, although the results are somewhat less convincing. Finally, the fusion of features (such as OTD and MFCCs) can achieve better discrimination of each performer than using any single timbre feature.

Based on the observations in this chapter, the timbre features do not model the performer's style very well on the SCC dataset, possibly due to the orchestral accompaniment introducing noise into the feature extraction process. Furthermore, due to the high time cost of the data annotation process and the unexpected noise generated by manual annotation, the performance of the method depends heavily on the quality of the data annotation, which leads to poor generalisation to other unlabelled datasets. Using any musical composition to identify performers without complex data annotation remains a problem. To address these issues, the next Chapter will present deep learning algorithms to acquire a more accurate and generalised approach for violinist identification.

Chapter 7

Transfer Learning for Violinist Identification

7.1 Introduction

In previous Chapters, the hand-crafted feature development and statistical modelling method are applied to describe the violinist's characteristic style, and the similarity among feature distributions are calculated to identify performers. Although such methods achieve good performance in violinist identification based on several datasets, it is highly likely that hand-crafted audio features cannot fully reflect the violinist's playing style, which also leads to suboptimal recognition results. In addition, the complicated note onset labelling is expensive in terms of time cost and results in poor generalisation on other unlabelled datasets. Therefore, we should explore other violinist identification (VID) methods to address these problems.

In recent years, deep neural networks (DNN) have been widely applied and shown great success in music information retrieval (MIR) research [44]. In music classification, instead of fitting a machine learning or statistical model using hand-crafted audio features, the deep learning approaches have

multiple trainable layers which learn the complicated relationship between the input and the output to predict the target label [44]. In addition, in polyphonic music like concerto or sonata, there is a risk that the accompaniment may bring noise to the hand-crafted audio features, so that they cannot accurately reflect the violinist's style. However, in a deep neural network, if the training and test data have the same distribution, the DNN can automatically learn features that reflect the violinist's style and use the learned features to identify the violinist. Furthermore, regarding the training phase of the DNN for recognising performers, the performer's ID is almost the only information that needs to be annotated, and no other complicated annotations are required. Therefore, we hypothesise that the deep learning method can work well in recognising performers and addressing the problem mentioned in the above paragraph.

However, although DNNs are very powerful in many MIR areas like music classification [207], music tagging [155], music emotion recognition [208] and music generation [209], there are not many works that use them to identify instrumentalists to our knowledge. One of the main reasons is that not many large-scale datasets contain performer information of each music piece. Existing music datasets like MagnaTagATune (MTAT) [210], the Million Song Dataset (MSD) [211], Jamendo [212] and GTZAN [213] are mostly designed for music tagging or classification. Moreover, training on a limited dataset tends to overfit the training set, leading to poor generalisation performance and unreliable results [214].

In order to solve the problem of insufficient datasets for training DNNs, the idea of transfer learning has been increasingly applied. Small datasets can train neural networks by transferring pre-trained weights, and achieve good performance in MIR tasks [176, 215, 216, 176, 217, 218]. Cramer [219] also found that pre-training a model on a large amount of data resulted in models that could be fine-tuned to downstream tasks with little data. Although the

deep learning method is rarely applied for identifying instrumentalists due to the lack of large-scale datasets, there are works in the related areas that can be considered source tasks. For example, the DNN trained for music auto-tagging [220, 145] is a reasonable source task due to many available large datasets published for this task, and its rich label set covers various aspects of music, e.g., genre, mood, era, and instrumentation. It is also considered a combination of multiple tasks such as genre classification, emotion recognition and instrument identification, which contributes to learning the relation between tags and audio content. In addition, DNN is applied for recognising individual styles of different singers from music performances, which is similar to instrumentalist identification. Some research has focused on singer recognition based on DNN [17, 18, 15], and some available large-scale datasets (e.g. artist20) are already published. Therefore singer recognition is regarded as another source task in our research. Since transfer learning can be used for different music classification and regression tasks [176], we hypothesise that pre-trained models for music tagging and singer identification (SID) can help identify instrumental players.

In this Chapter, we propose a method for violinist identification using the transfer learning method, which is based on pre-trained music tagging and singer identification models. We choose seven deep neural networks for music tagging and three neural networks for singer identification, then train them using corresponding source datasets to obtain pre-trained weights. Next, we retrain the selected models on two violin datasets (ASC, ACC) separately, and use pre-trained weights during initialisation. Details of the transfer learning method will be proposed in Section 7.2, and the results obtained from different pre-trained models and datasets are compared and discussed in Section 7.3.2.

7.2 Methods

In this section, the approach of violinist identification using transfer learning is presented. We first train seven music tagging models using three datasets: MSD, MTAT and Jamendo, respectively. Next, three singer identification models are trained using the artist20 dataset. These models are regarded as source tasks, which are introduced in Section 7.2.1. Then, we modify the model architecture and fine-tune the models on the ACC and ASC datasets separately, which will be presented in Section 7.2.2.

7.2.1 Source tasks

7.2.1.1 Music auto tagging

We select seven music tagging models as source task, including a fully convolutional network (FCN) [221], short-chunk CNN with Residual connections [220], Sample-level CNN [222], Musicnn [223], Harmonic CNN [224], Convolutional Recurrent Neural Network (CRNN) [145], and self-attention-based CRNN (self-attention-CRNN) [168]. The architecture of these models is introduced below.

The FCN consists of 4 convolutional layers and 4 max-pooling layers. It takes a log-amplitude Mel-spectrogram as input and predicts a 50-dimensional tag vector [221]. Similarly, another FCN with 7-layer CNN and a fully-connected layer and its extension with residual connections (named as short-chunk CNN) are validated in [220], which shows outstanding performance. Sample-level CNN [222] is an end-to-end model that takes raw audio waveforms as its inputs. It consists of ten 1D convolutional layers with 1×3 filters and 1×3 max-poolings, simpler and deeper than Mel spectrogram-based approaches [220]. Since a variation of Sample-level CNN with squeeze-and-excitation (SE) [225] blocks performs better than the original one, we use this model in our paper. Musicnn [223] is different from previously pro-

posed models, although it also uses Mel spectrograms as input. It is designed to rely on music domain knowledge. Harmonic CNN [224] takes advantage of trainable band-pass filters and harmonically stacked time-frequency representation inputs. The number of frequency bands is set to 128, and the number of harmonics is six.

Due to CRNN being widely used for music auto tagging [145], models like CRNN and self-attention-CRNN are taken into account. CRNN is a combination of CNNs and RNNs, where the CNN front-end extracts local features and the RNN back-end summarises them temporally. The architecture of self-attention-CRNN is similar to CRNN, and the only difference is that a self-attentive mechanism instead of RNN is used as temporal summarisation back-end [168]. The inputs of these two CRNN-based models are Mel-spectrograms.

We train these models using the MSD, MTAT and Jamendo datasets separately. All Mel-spectrogram-based approaches use 512-point FFT with a 50% overlap, and the frequency bins are all set as 128. Table 7.1 lists details of input for each model. When training these models, the audio resampling rate is set as 16000 Hz, which is kept the same as the original work [220] suggested. We also used an optimisation method that combines scheduled ADAM [226] and stochastic gradient descent (SGD) [227], which is also proposed in [168].

7.2.1.2 Singer Identification

Since the CRNN-based models have recently been used for singer identification and present good results [17, 18, 15], we select three CRNN-based models trained for singer identification as source tasks.

The first model is CRNNM [18], which extends original CRNN model [17] for SID. The input features of CRNNM are Mel-spectrogram and melody contour, where the melody contour is extracted using CREPE [54]. There

Table 7.1: The details of source tasks in our transfer learning experiment.

Task	Dataset	Models	Input Length	Input Feature	Classes
Music Tagging	MSD/ MTAT/ Jamendo	FCN	29.1s	Mel-spectrogram	50
		Musicnn	3s	Mel-spectrogram	50
		Harmonic CNN	5s	Stacked harmonic tensor	50
		Sample-level CNN	3.69s	Raw Waveform	50
		Short-chunk CNN	3.69s	Mel-spectrogram	50
		CRNN	29.1s	Mel-spectrogram	50
		CRNN-self-attention	15s	Mel-spectrogram	50
Singer Identification	Artist20	CRNNM	5s	Mel-spectrogram & Melody contour	20
		CRNN-attention	5s	Mel-spectrogram	20
		CRNN-attention-KNN	5s	Mel-Spectrogram	20

are two branches of CNN layers in the CRNNM model, each containing four convolutional layers. These two branches extract local timbre features based on two inputs separately, and the obtained feature maps are then concatenated together. Two GRU layers are followed to extract time-domain features on the concatenated features. Finally, a dense layer is applied to obtain desired singer label, where the cross entropy loss function is used.

Another two models, Attention-CRNN and Attention-CRNN-KNN are proposed in [15], which perform best in existing SID works using the artist20 dataset. Similar to the CRNN architecture in [17], there are also 4 CNN layers as well as 2 GRU layers. The CRNN model is followed by an attention layer and a dense layer in Attention-CRNN. Nevertheless, in Attention-CRNN-KNN, the CRNN model is followed by an attention layer and a KNN classifier.

Therefore, we train CRNNM, Attention-CRNN and Attention-CRNN-KNN using the artist20 dataset, following the same training setup (i.e., data split method, filters numbers, kernel sizes, optimiser, learning rate, activation functions, loss function) described in the original works. The audio input length of each model is set as 5s, and Mel-spectrogram bins are all set as 128, using 512-point FFT with a 50% overlap. We use ADAM for learning rate control and cross entropy is applied as a loss function. Meanwhile, the

early stopping [228] is also applied, which is the same as suggested by [145].

7.2.2 Target tasks

After training all models in the source task on source datasets, the weight of each layer in pre-trained models is obtained. In order to adapt the demand of the target task on different violin datasets, we change the final dense layer of each pre-trained model, which outputs probabilities of violinists instead of original labels or singers. Next, we retrain these models separately using weights from each pre-trained model during initialisation, and the best model is selected based on validation loss. Finally, the violinist classification results from each model can be obtained.

Since music tagging is a multi-label classification task, the Binary Cross Entropy Loss is used as loss function in original models [222]. However, the performer identification is a single-label classification task (in our case), and cross entropy loss is suitable and widely used in similar tasks [229]. Thus we use cross entropy as loss function to retrain all music tagging models on violin datasets. As the singer identification is also a single-label classification task, the loss function is cross entropy, which does not need to be changed further. Nevertheless, the dimension of output labels of each pre-trained model should be matched with the number of violinists in each dataset.

As an example, the transfer learning process using a pre-trained Musicnn model to identify violinists on the ACC dataset is shown in Figure 7.1. We transfer the learned knowledge from the pre-trained music tagging network, then modify the output layer and fine-tune the model using ACC data to obtain violinist identification results.

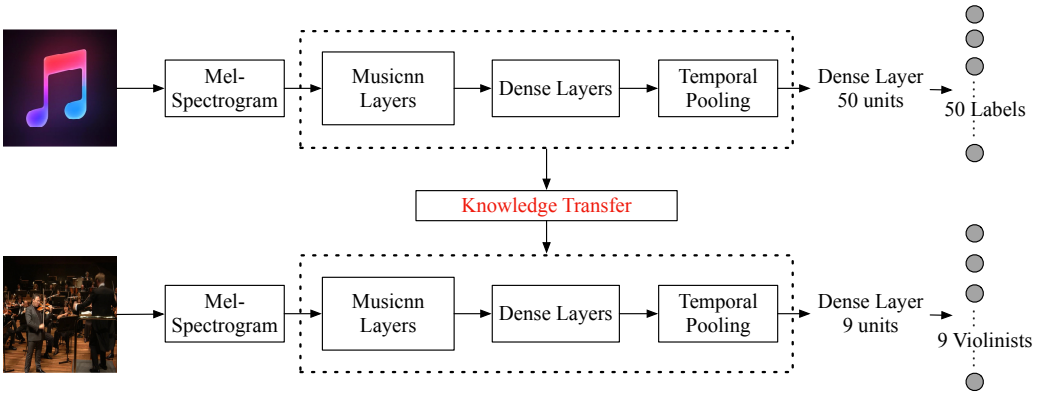


Figure 7.1: Transfer learning process using pre-trained Musicnn model on ACC dataset.

7.3 Experiments

In this section, we first introduce the data preparation, including data split and segmentation procedures. Then we present implementation details of the experiment in Section 7.3.1.2. Finally, we show the results in Section 7.3.2.

7.3.1 Experimental Setup

7.3.1.1 Dataset preparation

In the ACC dataset, to decrease the influence of orchestra accompaniment, original concerto recordings are first segmented into several short clips, introduced in Section 3.2.4. However, to adapt the requirement of input length for each model, the pre-segmented audio clips must be further divided into different lengths. To adapt the requirement of different input lengths for each model, we segment the audio for each performer in lengths of 29.1s, 15s, 5s, 3.69s and 3s separately without overlaps after re-sampling the audio using $F_s = 16000$Hz, which is kept same as we did in the training phase of all source tasks. For all audio segments of each performer, we randomly shuffle them into training, validation and test sets using a ratio of 6:2:2, which avoids an unbalanced amount of data for each performer in different sets.

7.3.1.2 Estimation metric and baseline method

In order to evaluate and compare the violinist identification performance of the proposed model, the macro F1-score is used as the evaluation metric. To validate the effectiveness of our proposed transfer learning method and compare its performance with the methods in previous chapters, we consider the results in Chapter 4-6 as baselines.

7.3.2 Results

Table 7.2 and Table 7.3 summarise the results obtained by our proposed method, with the test F1-score based on the ACC and ASC datasets, respectively. To compare the differences in results with and without transfer learning, we first show the results trained from scratch (using random initialisation) for each model, corresponding to the "Scratch" column in each Table. The evaluation of violinist identification based on different source datasets and pre-trained models is then shown separately. To make the results can be observed intuitively, we also show them in Figure 7.2. The top bar chart shows the result using the ACC dataset, while the bottom shows the F1-score using the ASC dataset.

Table 7.2: Violinist identification results using ACC dataset.

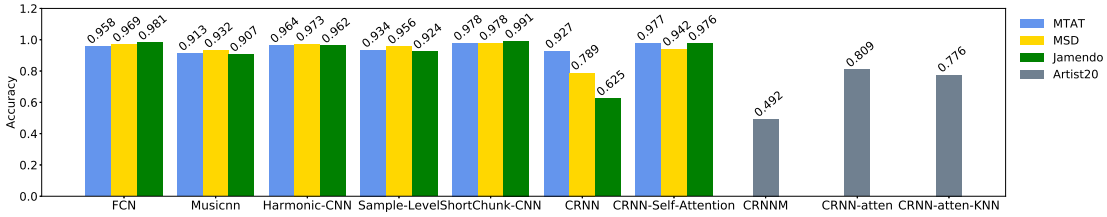
Models	Scratch	MTAT	MSD	Jamendo	Artist20
FCN	0.950	0.958	0.969	0.981	–
Musicnn	0.905	0.913	0.932	0.907	–
Harmonic CNN	0.955	0.964	0.973	0.962	–
Sample-level CNN	0.908	0.934	0.956	0.924	–
Short-chunk CNN	0.976	0.978	0.978	**0.991**	–
CRNN	0.548	0.927	0.789	0.625	–
CRNN-self-attention	0.937	0.977	0.942	0.976	–
CRNNM	0.479	–	–	–	0.492
CRNN-attention	0.755	–	–	–	0.809
CRNN-attention-KNN	0.745	–	–	–	0.776

Table 7.3: Violinist identification results using ASC dataset.

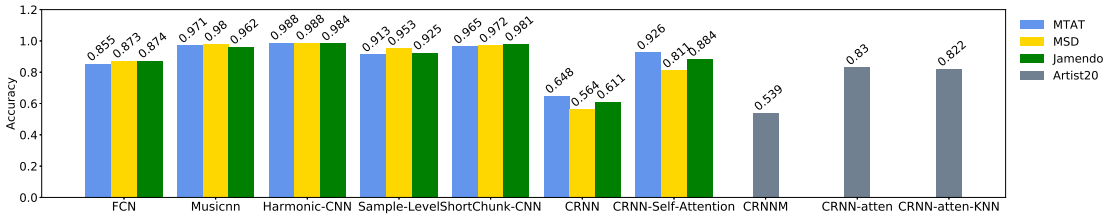
Models	Scratch	MTAT	MSD	Jamendo	Artist20
FCN	0.850	0.855	0.873	0.874	–
Musicnn	0.960	0.971	0.980	0.962	–
Harmonic CNN	0.981	**0.988**	**0.988**	0.984	–
Sample-level CNN	0.786	0.913	0.953	0.925	–
Short-chunk CNN	0.953	0.965	0.972	0.981	–
CRNN	0.513	0.648	0.564	0.611	–
CRNN-self-attention	0.978	0.926	0.811	0.884	–
CRNNM	0.546	–	–	–	0.539
CRNN-attention	0.825	–	–	–	0.830
CRNN-attention-KNN	0.793	–	–	–	0.822

It can be seen in the Table that knowledge transfer is beneficial for improving violinist identification performance. Short-chunk CNN and Harmonic CNN showed the best results for both target datasets, no matter which source datasets were used for pre-training. The Short-chunk CNN obtains the best F1-score on the ACC dataset pre-trained on the Jamendo dataset, which is 0.991; the best solo violinists identification performance is 0.988 in F1-score, which is obtained by Harmonic CNN pre-trained on the MSD dataset.

For the CRNN models, self-attention mechanisms can improve the performance, no matter which pre-trained model is used. It can be observed that the results obtained in Table 7.3 show that "Scratch" outperforms other source datasets. It may be because the attention layer can capture the characteristic style of violin solo performers when training from scratch, while the pre-trained weights do not help identify those performers due to the gap between the source dataset and target dataset. However, the results obtained from pre-trained SID networks are generally inferior. One possible reason is that the characteristic features of singers are not directly transferable to identify the results obtained from pre-trained SID networks. CRNNM performs worst no matter which target dataset is trained on, which denotes its input branch of the main melody contour may bring noise to the violinist



(a) Results based on ACC dataset.



(b) Results based on ASC dataset.

Figure 7.2: Violinist identification results based on two violin datasets using different pre-trained models and source datasets.

identification in our case.

In previous chapters, based on datasets constructed from solo recordings, Chapter 5 reports a 0.94 in F1-score when using MFCCs. In addition, based on datasets built from concerto recordings, the best F1-score is 0.956, which was observed using the IVN dataset with seven fused audio features, and 0.797 in the F1-score is found when using the OTD_AVG feature on the SCC dataset. It is easy to find that the best results obtained from the previous chapters are worse than those obtained with the transfer learning method, which indicates it is highly effective for identifying violinists.

7.3.3 Discussion

In general, the pre-trained music tagging networks perform better than SID models. It is probably because the former models were pre-trained on datasets containing a broader set of musical styles, and the models were designed to facilitate the output of 50 broad music labels belonging to different categories.

Therefore music tagging models can learn more detailed musical features, which may include feature spaces suitable for characterising violinists' styles. In contrast, the source dataset of the latter task (artist20) contains human voices, and the corresponding SID models were designed to find stylistic features of vocal performances, which is somewhat different from our target task.

Among the pre-trained music tagging results, models trained on shorter music clips (short-chunk CNN, sample-level CNN, harmonic CNN and Musicnn) outperformed models trained on longer music clips (FCN, CRNN). Intuitively, when the models are trained on short inputs, there are a larger number of samples during the training process, and the performer's style can likely be identified within a few seconds, which brings good performance to these models. Moreover, FCN and CRNN perform better on the ACC dataset than the ASC dataset, suggesting that when the data amount is relatively small, longer input brings a smaller number of samples, leading to weaker results.

Finally, comparing transfer learning with methods proposed in previous Chapters, it can be found that transfer learning performs much better in VID results. There may be several reasons for this. First, the features learned in deep learning layers are intended to fit the output better, providing a more comprehensive representation of the player's style in the audio signal than hand-crafted audio features. Second, due to the limited size of the violin dataset we constructed and the fact that some players have similar playing habits and preferences, the distribution of hand-crafted features is very similar between some players, which makes these features less discriminatory. In transfer learning, however, the weights obtained from pre-trained models on large-scale source datasets are already good at extracting basic audio features, and subsequent retraining can focus more on the differences between performers, making more effective use of the dataset.

Although we used multiple source datasets to train different source tasks, the pre-trained neural network can identify violinists based on different music scenarios, and the amount of data used to fine-tune the model does not need to be large. This dramatically improves the generalisation capability of the method, without the need to design different audio features to adjust different scenarios. Also, the method does not require complicated note onset labels, which reduces much workload in data annotation and pre-processing.

However, transfer learning also has its shortcomings. First, it is not easy to analytically explore and understand which specific perceptual musical features work well to distinguish violinists. All deep learning models applied in this Thesis are end-to-end, and all feature extraction and learning processes are implemented in the hidden layers. Thus it is not easy to gain a new understanding of music theory on performers' styles. In addition, the training process is time-consuming, both in the training phase of source tasks and target tasks. In particular, training needs to be restarted when the model is modified, which significantly increases the time cost.

Furthermore, although the fine-tuning of transfer learning does not require much data, it still requires a certain amount of data to retrain the models, which limits its application for some conditions. For example, it is not feasible to identify performers based on only 20 individual notes in a concerto movement, but the method proposed in Chapter 4 makes it possible. Finally, although transfer learning can avoid overfitting to some extent, there is still a risk of overfitting when fine-tuning the neural network on a small dataset. To address this issue, one possible approach would be to evaluate the performance of pre-trained neural networks using an utterly new violinist dataset. If the results are also reasonably well, it would indicate that the overfitting is not severe. However, as there is no publicly available dataset, this work will be placed in future work.

7.4 Summary

In this Chapter, we propose a transfer learning-based method for violinist identification. First, several deep neural networks applied to music tagging and singer identification are regarded as source tasks, and their architecture and basic parameter settings are introduced in Section 7.2.1. These models are first trained separately based on their corresponding source datasets, and the pre-training implementation is then presented in Section 7.2.1. Next, we modify the structure of these models to suit the requirement of violinist identification based on our datasets, details of which are illustrated in Section 7.3. Then, these pre-trained models are fine-tuned on ASC and ACC datasets, and the results of violinist identification are obtained and discussed in Section 7.3.2. The results show that the pre-trained models can be successfully adapted to the target task and exceed the previous Chapters' methods, achieving high performance on both datasets. Finally, the advantages and shortcomings of the transfer learning method are discussed in Section 7.3.3.

Chapter 8

Conclusion

This Thesis presents the research process involved in designing and evaluating violinist identification (VID) approaches. This Chapter summarises the main contributions and draws fundamental conclusions from the system design and experiments described throughout the Thesis. The possible future development of this research is also included in the end.

8.1 Summary of Contributions

The significant contributions of this Thesis can be summarised in four aspects: i) novel violinist datasets construction; ii) audio features development for describing violinist's playing style; iii) comprehensive statistical algorithms for modelling and identifying violinist's style; iv) a transfer learning approach for violinist identification.

8.1.1 Dataset Construction

Most existing violin datasets are constructed for music expression analysis or instrument recognition, whereas the performer information is rarely included. In Chapter 3, we constructed three concerto datasets from concerto perfor-

mances recorded by nine famous violinists, which is applied for evaluating our proposed VID methods on master performers' performance. First, vibrato notes are manually segmented to compose an "Isolated Vibrato Notes" (IVN) dataset by labelling their onset and offset positions, which contains 250 notes for each performer. Next, to avoid the influence of orchestra accompaniment, we remove the parts of the music without violin (e.g. prelude, interlude) or where the violin cannot be heard clearly. The remaining music clips are included to form the "all concerto clips" (ACC) dataset. It is used to train deep learning models in Chapter 7. Finally, we selected some clips from the ACC dataset and annotated the onset positions of each note. The selected music clips, as well as the annotations, are taken to constitute the "selected concerto clips" (SCC) dataset, and it is applied for assessing the proposed methods in Chapter 6.

In addition, to further investigate the effectiveness of designed VID methods and provide a solo music-based scenario for the VID, we built two datasets from solo musical scale recordings from the Bilbao project. A "selected solo clips" (SSC) dataset consists of scales played by ten selected performers, annotating each note's onset time. This dataset was then used in Chapter 5 to measure the effectiveness of timbre features in describing a violinist's style. In addition, all scales in the original recordings played by the 22 performers were manually labelled with the performer's label, which was named "all solo clips" (ASC) dataset. This dataset is used to train deep learning models, as described in Chapter 7.

Establishing the above dataset provides a basis for assessing the proposed violinist identification methods. It can also be used for other relevant tasks (e.g. violin style analysis, style transfer) in the MIR community.

8.1.2 Audio Feature Development

We designed and extracted three categories of note-level audio features to investigate which audio factors can describe a violinist's style. In Chapter 4, as vibrato plays an essential role in violin playing, four features have been developed to describe the habits and characteristics of the performer's vibrato: *Average Vibrato Extent*, *Average Vibrato Rate*, *Standard Deviation of Vibrato Extent* and *Standard Deviation of Vibrato Rate*. All vibrato features are extracted based on the main melody of the note, which is obtained using the "MELODIA" algorithm.

In addition to the vibrato features which reflect the characteristics of left-hand playing, six note-level timbre features (including *Spectral Centroid*, *Spectral Bandwidth*, *Spectral Contrast*, *RMS Energy*, *MFCCs*, *Zero-Crossing Rate*) have been designed to describe the right-hand playing habits and preferences. However, the raw timbre features are not only influenced by the performer's interpretation, but also by recording conditions, instruments or other factors. To address these problems, we extract standardised features at note level to model the timbre variations in each note, rather than the original feature values.

In addition, in Chapter 5, to explore whether timbre features are influenced by instrument, we design two experiments based on the *SSC* dataset, including violinist identification as well as violin identification, respectively. The results show that those features do not help identify violins, but they perform very well in identifying violinists, further validating the effectiveness and robustness of the designed timbre features for identifying violinists.

Finally, in Chapter 6, we developed two features to represent performer's expressive timing characteristic, including *Onset Time Deviation* (OTD) and *Note Duration* (ND). Corresponding feature extraction methods and comparing these features are also proposed in Chapter 6.

8.1.3 Violinist Identification using Statistical Distributions

We propose a violinist identification method based on similarity calculation among audio feature distributions. For each violinist, the global distribution of audio features is considered to describe the performer's style. Then, the similarity between feature distributions of different performers can be calculated and regarded as violinist identification results. In Chapter 4, three distributions (including Histogram, Kernel Density Estimation (KDE), and Gaussian Mixture Model (GMM)) are applied, and methods of violinist identification based on each model are presented and compared. The results show that the distribution selection does not significantly affect the results, while the KDE performs best on most vibrato features, while the histogram performs best on most timbre features. In addition, these methods are also compared with standard machine learning models (e.g. KNN, SVM), which are frequently used for similar MIR tasks, which suggests our proposed method performs better than those models.

The VID performance of each feature is also compared. In Chapter 4, we attempt to identify violinists based on vibrato features and timbre features, which are evaluated using the IVN dataset. Results show that *Average Vibrato Rate* is the best vibrato feature to distinguish violinists, whereas *Standard Deviation of Vibrato Extent* performs the worst. In addition, among timbre features, the *MFCCs* can identify violinists well (F1-score is 0.865), while *ZCR* is not very helpful in distinguishing violinists. Next, in Chapter 5, we evaluate the timbre features on the SSC dataset to validate their effectiveness for identifying violin players, where 0.941 and 0.917 (F1-score) are given based on MFCCs and Spectral Contrast, and the worst is ZCR as well. However, the results of violin identification with these features are poor, approximately the same as the random baseline, indicating the features

cannot represent the violin's property well. Finally, the VID based on short music clips is achieved in Chapter 6, using timing and timbre features on the SCC dataset. The observed results show that the 0.797 in F1-score was found using the OTD_AVG, which is better than using any other single feature on the same dataset. However, the timbre features perform much worse on the SCC dataset than on the IVN and SSC dataset, which indicates that the accompaniment may greatly influence such features' violinist style modelling capacity.

Furthermore, we present a method to fuse different features to identify violinists. It is observed that feature fusion helps to improve the discrimination of some individual performers significantly, but the overall VID results are not improved.

8.1.4 Transfer Learning for Violinist Identification

In order to improve the generalisation of the VID method and to further improve the accuracy of violinist identification, we propose a deep learning approach in Chapter 7. Due to the limited amount of data, transfer learning is considered. The source tasks are music tagging and singer identification. Seven CNN-based models are trained for music tagging on datasets including MSD, MTAT and Jamendo separately, and these models are then retrained using weights from each pre-trained model during initialisation. A similar mechanism can be found when using singer identification as source task, where three models are pre-trained on the Artist20 dataset, and the trained models are then fine-tuned for violinist identification. Finally, the violinist classification results obtained for each model are compared. The best result on ACC dataset is obtained by the Short chunk CNN pre-trained on the Jamendo dataset, which is 0.991; the best performance on the ASC dataset is 0.988, which is obtained by Harmonic CNN pre-trained on the MSD dataset. The VID performances are significantly improved compared to the methods

proposed above.

The transfer learning method gives better results than the methods mentioned in the previous Chapters and does not require complicated data pre-processing processes. However, as we have discussed before, the method is not very interpretable, and it is unclear what valuable features are 'learned' by the neural networks. Therefore its contribution to the development of musicology is limited.

8.2 Future Perspectives

8.2.1 Dataset Optimisation

As the author creates the datasets used in this Thesis, the amount of data is limited due to the massive workload of data annotation. In the future, we can try to enlarge the dataset by labelling more performers' recordings and verify the reliability of our proposed method on the larger dataset. In addition, in Chapter 6, it is evident that the timbre features are less able to discriminate violinists based on the concerto recordings than on the solo recordings. This is primarily due to the influence of the accompaniment in the concerto, which introduces noise into the extracted timbre features, making them not solely represent the timbre variations of the violinist's playing. To avoid the influence of accompaniment, we may apply source separation to isolate the violin performance from polyphonic music in the future, and the audio features and distributions would potentially model the performers' individual playing styles better.

In addition, the data pre-processing applied in Chapter 4, 5 and 6 are not automatic; the loudness normalisation, zero-phase filtering and silence removal are done separately, which require complicated parameter tuning and limits the application to other tasks. In the future, an automatic data pre-processing method for multiple scenarios should be proposed, such as

automatic detection of silences and automatic adjustment of filter parameters. Integrating these pre-processing modules into a whole will help optimise machine learning models' subsequent optimisation and selection.

8.2.2 New Features Exploration

By combining multiple audio descriptors, hand-crafted features are used to describe different aspects of the music (e.g., vibrato, timbre, timing), which can objectively represent the violinist's style. An intuitive future direction is to develop new audio features to characterise violinist's playing better. In addition to the design of hand-crafted features, feature learning methods can be applied, which may allow more information relevant to the style of performance can be captured from the audio signal.

Furthermore, the features presented in this Thesis are at note level, which is chosen because it balances the complexity of the analysis with the integrity of musical expression. However, higher level features (such as bar level or beat level) are also essential to fully characterise the performer's performance. In future work, we intend to extract different hierarchical audio features to identify violinists based on musical structure analysis.

8.2.3 Model Optimisation

We present a violinist identification approach based on feature distributions and similarity calculation. Although this method performs better than commonly used machine learning models (like KNN and SVM), the results based on the SCC dataset are not convincing, where the F-score metrics are all below 0.9. Further research could be carried out using more complicated statistical models to obtain audio feature distributions, enabling better identification of violinists.

In addition, ten deep neural networks pre-trained for other MIR tasks are

considered source tasks, then fine-tuned on our proposed violin datasets to identify violinists. In order to explore the effectiveness of transfer learning, the basic structure of these models is largely retained, and no extensive modifications are applied to them. In the future, the model architecture could be optimised to obtain better performance (e.g. the number of layers, the loss function or kernel design), and we can evaluate the models' performance on the new violin dataset.

8.2.4 Performer Identification for Wider Scenarios

The violinist identification methods explored in this Thesis have been evaluated on known performers and compositions in the dataset, but their validity has not been verified for unknown musical pieces and performers. In further research, we could first apply the proposed method to unknown performances played by known violinists. Alternatively, we could add more performers and corresponding performances to the dataset and evaluate the effectiveness of different audio features and classifiers based on a larger dataset.

On the other hand, the methods proposed in this Thesis aim to identify violinists, but they can identify performers who play other instruments. For example, the developed vibrato features could be attempted to characterise viola players, cellists or performers playing other stringed instruments. Moreover, in Chapter 7, the pre-trained models are fine-tuned on our violinist datasets and show promising results for violinist identification. If these pre-trained models were retrained on other datasets (e.g., piano or flute dataset), would they be able to classify pianists or flute players? This remains a question and worth to be explored in the future.

Appendix A

Dataset Details

As mentioned in Chpater 3, we constructed two groups of datasets to evaluate violinist identification algorithms proposed in this Thesis. Although Table 3.1 shows the selection of concertos, the specific information about the albums is not explained in detail. To give readers a clearer picture of our research and to make it easier to reproduce our work, the information of each album is presented in Table A.1, including the performer name, repertoire, ASIN code, and original release date. Since all albums are bought from Amazon, the specific CD album can be reached from https://amazon-asin.com/ with the corresponding ASIN code.

Table A.1: The details of selected CD albums.

Performer	Manufacturer	Repertoire	ASIN Code	Date
Accardo	Decca	Beethoven Violin Concerto Op.61	B01JTRQ1N4	2016
Accardo	Decca	Brahms Violin Concerto Op.77	B01KHURU5U	2016
Accardo	Decca	Mendelssohn Violin Concerto Op.64	B01KHUQ7W2	2016
Accardo	Philips	Sibelius Violin Concerto Op.47	B01M6YE2XM	2016
Accardo	Philips	Tchaikovsky Violin Concerto Op.35	B01M6YE2XM	2016
Heifetz	SONY	Beethoven Violin Concerto Op.61	B006XOBFHO	2012
Heifetz	SONY	Brahms Violin Concerto Op.77	B00E00GXWA	2003
Heifetz	SONY	Mendelssohn Violin Concerto Op.64	B00E00GXWA	2003
Heifetz	EMI Classics	Sibelius Violin Concerto Op.47	B000002S2U	1991
Heifetz	SONY	Tchaikovsky Violin Concerto Op.35	B00E00GXWA	2003
Menuhin	Documents	Beethoven Violin Concerto Op.61	B0113A5ASC	2015
Menuhin	Documents	Brahms Violin Concerto Op.77	B0113A5ASC	2015
Menuhin	Deutsche Grammophon	Mendelssohn Violin Concerto Op.64	B00R74MXP2	2015
Menuhin	Documents	Sibelius Violin Concerto Op.47	B0113A5ASC	2015
Menuhin	Deutsche Grammophon	Tchaikovsky Violin Concerto Op.35	B00R74MXP2	2015
Mutter	Deutsche Grammophon	Beethoven Violin Concerto Op.61	B000W99IKW	2007
Mutter	Deutsche Grammophon	Brahms Violin Concerto Op.77	B000001GNG	1995
Mutter	Deutsche Grammophon	Mendelssohn Violin Concerto Op.64	B000001GNG	1995
Mutter	Deutsche Grammophon	Sibelius Violin Concerto Op.47	B000001GRK	1996
Mutter	Deutsche Grammophon	Tchaikovsky Violin Concerto Op.35	B0002U9G7G	2004
Oistrakh	Deutsche Grammophon	Beethoven Violin Concerto Op.61	B003GW1P1C	2010
Oistrakh	Deutsche Grammophon	Brahms Violin Concerto Op.77	B000001GQI	1996
Oistrakh	Naxos Historical	Mendelssohn Violin Concerto Op.64	B000M2DNU0	2007
Oistrakh	SONY	Sibelius Violin Concerto Op.47	B004NSHBCK	2014
Oistrakh	Deutsche Grammophon	Tchaikovsky Violin Concerto Op.35	B000001GQI	1996
Perlman	Warner Classics	Beethoven Violin Concerto Op.61	B010FULJVS	2015
Perlman	EMI Classics	Brahms Violin Concerto Op.77	B0000AF1LM	1992
Perlman	Warner Classics	Mendelssohn Violin Concerto Op.64	B010DUTTLC	1992
Perlman	Red Seal	Sibelius Violin Concerto Op.47	B0001TSWMI	2004
Perlman	Red Seal	Tchaikovsky Violin Concerto Op.35	B0001TSWMI	2004
Stern	SONY	Beethoven Violin Concerto Op.61	B000026QSW	1993
Stern	SONY	Brahms Violin Concerto Op.77	B00KV192M0	2014
Stern	SONY	Mendelssohn Violin Concerto Op.64	B0000025JL	1990
Stern	SONY	Sibelius Violin Concerto Op.47	B000002AXW	2012
Stern	SONY	Tchaikovsky Violin Concerto Op.35	B000002AXW	2012
Vengerov	EMI Classics	Beethoven Violin Concerto Op.61	B000B63IDO	2005
Vengerov	TELDEC	Brahms Violin Concerto Op.77	B00000HZOB	1999
Vengerov	Warner Classics	Mendelssohn Violin Concerto Op.64	B00006IWQ8	2002
Vengerov	TELDEC	Sibelius Violin Concerto Op.47	B005CNKTCO	2011
Vengerov	Warner Classics	Tchaikovsky Violin Concerto Op.35	B00006IWQ8	2002
Zukerman	Decca	Beethoven Violin Concerto Op.61	B000025KBO	2008
Zukerman	Deutsche Grammophon	Brahms Violin Concerto Op.77	B00LWIK8ZQ	2014
Zukerman	Decca	Mendelssohn Violin Concerto Op.64	B000025KBO	2008
Zukerman	Deutsche Grammophon	Sibelius Violin Concerto Op.47	B00LWIK8ZQ	2014
Zukerman	Decca	Tchaikovsky Violin Concerto Op.35	B000025KBO	2008

Bibliography

[1] Sandrine Vieillard, Mathieu Roy, and Isabelle Peretz. Expressiveness in musical emotions. *Psychological research*, 76(5):641–653, 2011.

[2] Patrik N Juslin and Petri Laukka. Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological bulletin*, 129(5):770, 2003.

[3] Mathieu Barthet, Philippe Depalle, Richard Kronland-Martinet, and Søølvi Ystad. Analysis-by-synthesis of timbre, timing, and dynamics in expressive clarinet performance. *Music perception: An interdisciplinary journal*, 28(3):265–278, 2011.

[4] Bruno H Repp. Diversity and commonality in music performance: An analysis of timing microstructure in schumann's "träumerei". *The Journal of the Acoustical Society of America*, 92(5):2546–2568, 1992.

[5] Efstathios Stamatatos. Quantifying the differences between music performers: Score vs. norm. In *ICMC*. Citeseer, 2002.

[6] Roberto Bresin and Giovanni Umberto Battel. Articulation strategies in expressive piano performance analysis of legato, staccato, and repeated notes in performances of the andante movement of mozart's sonata in g major (k 545). *Journal of New Music Research*, 29(3):211–224, 2000.

[7] Efstathios Stamatatos and Gerhard Widmer. Automatic identification of music performers with learning ensembles. *Artificial Intelligence*, 165(1):37–56, 2005.

[8] Werner Goebl. Skilled piano performance: Melody lead caused by dynamic differentiation. In *Proc. of the 6th Int. Conf. on Music Perception and Cognition*. Citeseer, 2000.

[9] Craig Saunders, David R Hardoon, John Shawe-Taylor, and Gerhard Widmer. Using string kernels to identify famous performers from their playing style. In *European Conference on Machine Learning*, pages 384–395. Springer, 2004.

[10] Rafael Ramirez, Esteban Maestre, Antonio Pertusa, Emilia Gomez, and Xavier Serra. Performance-based interpreter identification in saxophone audio recordings. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3):356–364, 2007.

[11] Nadine Kroher and Emilia Gómez. Automatic singer identification for improvisational styles based on vibrato, timbre and statistical performance descriptors. In *ICMC*, 2014.

[12] Tong Zhang. Automatic singer identification. In *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, volume 1, pages I–33. IEEE, 2003.

[13] Hamid Eghbal-Zadeh, Bernhard Lehner, Markus Schedl, and Gerhard Widmer. I-vectors for timbre-based music similarity and music artist classification. In *ISMIR*, pages 554–560, 2015.

[14] Deepali Y Loni and Shaila Subbaraman. Timbre-vibrato model for singer identification. In *Information and Communication Technology for Intelligent Systems*, pages 279–292. Springer, 2019.

[15] Xulong Zhang, Jiale Qian, Yi Yu, Yifu Sun, and Wei Li. Singer identification using deep timbre feature learning with knn-net. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3380–3384. IEEE, 2021.

[16] Daniel PW Ellis. Classifying music audio with timbral and chroma features. 2007.

[17] Zain Nasrullah and Yue Zhao. Music artist classification with convolutional recurrent neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[18] Tsung-Han Hsieh, Kai-Hsiang Cheng, Zhe-Cheng Fan, Yu-Ching Yang, and Yi-Hsuan Yang. Addressing the confounds of accompaniments in singer identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020.

[19] Bruno Gingras, Tamara Lagrandeur-Ponce, Bruno L Giordano, and Stephen McAdams. Perceiving musical individuality: performer identification is dependent on performer expertise and expressiveness, but not on listener expertise. *Perception*, 40(10):1206–1220, 2011.

[20] Rüdiger Flach, Günther Knoblich, and Wolfgang Prinz. Recognizing one's own clapping: the role of temporal cues. *Psychological research*, 69(1):147–156, 2004.

[21] Pei-Ching Li, Li Su, Yi-hsuan Yang, Alvin WY Su, et al. Analysis of expressive musical terms in violin using score-informed and expression-based audio features. In *ISMIR*, pages 809–815, 2015.

[22] Rafael Ramirez, Esteban Maestre, Alfonso Perez, and Xavier Serra. Automatic performer identification in celtic violin audio recordings. *Journal of New Music Research*, 40(2):165–174, 2011.

[23] Miguel Molina-Solana, Josep Lluís Arcos, and Emilia Gomez. Identifying violin performers by their expressive trends. *Intelligent Data Analysis*, 14(5):555–571, 2010.

[24] Chi-Ching Shih, Pei-Ching Li, Yi-Ju Lin, AWY Su, L Su, and YH Yang. Analysis and synthesis of the violin playing styles of heifetz and oistrakh. In *Proc. Int. Conf. Digital Audio Effects*.

[25] Jonna K Vuoskoski and Tuomas Eerola. The role of mood and personality in the perception of emotions represented by music. *Cortex*, 47(9):1099–1106, 2011.

[26] Caroline Palmer. Music performance. *Annual review of psychology*, 48(1):115–138, 1997.

[27] Tuomas Eerola, Anders Friberg, and Roberto Bresin. Emotional expression in music: contribution, linearity, and additivity of primary musical cues. *Frontiers in psychology*, 4:487, 2013.

[28] Heinrich Schenker and Felix Salzer. *Five Graphic Music Analyses (F nf Urlinie-Tafeln)*. Courier Corporation, 1969.

[29] Dirk Vorberg and Rolf Hambuch. On the temporal control of rhythmic performance. *Attention and performance VII*, pages 535–555, 1978.

[30] Alicia Fernández-Sotos, Antonio Fernández-Caballero, and José M Latorre. Influence of tempo and rhythmic unit in musical emotion regulation. *Frontiers in computational neuroscience*, 10:80, 2016.

[31] Anne J Blood, Robert J Zatorre, Patrick Bermudez, and Alan C Evans. Emotional responses to pleasant and unpleasant music correlate with activity in paralimbic brain regions. *Nature neuroscience*, 2(4):382, 1999.

[32] Di Sheng. *Intelligent Control of Dynamic Range Compressor*. PhD thesis, Queen Mary University of London, 2020.

[33] Alf Gabrielsson. The performance of music. In *The psychology of music*, pages 501–602. Elsevier, 1999.

[34] Jae Won Noella Jung. Jascha heifetz, david oistrakh, joseph szigeti: Their contributions to the violin repertoire of the twentieth century. 2007.

[35] Christopher Dobrian and Daniel Koppelman. The 'e'in nime: musical expression with new computer interfaces. 2006.

[36] Laurel S Pardue. *Violin augmentation techniques for learning assistance*. PhD thesis, Queen Mary University of London, 2017.

[37] John M Geringer, Michael L Allen, and Rebecca B Macleod. String vibrato: Research related to performance and perception. *String Research Journal*, 1(1):7–23, 2010.

[38] Joel Berman, Barbara Garvey Jackson, and Kenneth Sarch. *Dictionary of Bowing and Pizzicato terms*. American String Teachers Association with the National School Orchestra ..., 1999.

[39] Alfonso Perez-Carrillo. Violin timbre navigator: Real-time visual feedback of violin bowing based on audio analysis and machine learning. In *International Conference on Multimedia Modeling*, pages 182–193. Springer, 2019.

[40] Ivan Galamian and Sally Thomas. *Principles of violin playing and teaching*. Courier Corporation, 2013.

[41] Susan Kempter. *How muscles learn: Teaching the violin with the body in mind*. Alfred Music Publishing, 2003.

[42] Mi Tian. *A Cross-Cultural Analysis of Music Structure*. PhD thesis, Queen Mary University of London, 2016.

[43] Carol L Krumhansl. Rhythm and pitch in music cognition. *Psychological bulletin*, 126(1):159, 2000.

[44] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark Sandler. A tutorial on deep learning for music information retrieval. *arXiv preprint arXiv:1709.04396*, 2017.

[45] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-hill New York, 1986.

[46] Ervin Sejdić, Igor Djurović, and Jin Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital signal processing*, 19(1):153–183, 2009.

[47] Douglas O'shaughnessy. *Speech communications: Human and machine (IEEE)*. Universities press, 1987.

[48] Christopher J Plack, Andrew J Oxenham, and Richard R Fay. *Pitch: neural coding and perception*, volume 24. Springer Science & Business Media, 2006.

[49] Josh H McDermott and Andrew J Oxenham. Music perception, pitch, and the auditory system. *Current opinion in neurobiology*, 18(4):452–463, 2008.

[50] Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[51] Matthias Mauch and Simon Dixon. pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663. IEEE, 2014.

[52] Justin Salamon, Emilia Gómez, Daniel PW Ellis, and Gaël Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014.

[53] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[54] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165. IEEE, 2018.

[55] François Rigaud and Mathieu Radenen. Singing voice melody transcription using deep neural networks. In *ISMIR*, pages 737–743, 2016.

[56] Rachel M Bittner, Brian McFee, and Juan P Bello. Multitask learning for fundamental frequency estimation in music. *arXiv preprint arXiv:1809.00381*, 2018.

[57] Sebastian Böck and Markus Schedl. Polyphonic piano note transcription with recurrent neural networks. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 121–124. IEEE, 2012.

[58] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

[59] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, 2016.

[60] Ioannis Karydis. Symbolic music genre classification based on note pitch and duration. In *East European Conference on Advances in Databases and Information Systems*, pages 329–338. Springer, 2006.

[61] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.

[62] Caroline Palmer and Sean Hutchins. What is musical prosody? *Psychology of learning and motivation*, 46:245–278, 2006.

[63] Maureen Mellody and Gregory H Wakefield. The time-frequency characteristics of violin vibrato: Modal distribution analysis and synthesis. *The Journal of the Acoustical Society of America*, 107(1):598–611, 2000.

[64] Eric Prame. Measurements of the vibrato rate of ten singers. *The journal of the Acoustical Society of America*, 96(4):1979–1984, 1994.

[65] Hee-Suk Pang and Doe-Hyun Yoon. Automatic detection of vibrato in monophonic music. *Pattern Recognition*, 38(7):1135–1138, 2005.

[66] Luwei Yang, Elaine Chew, and Sayid-Khalid Rajab. Vibrato performance style: A case study comparing erhu and violin. 2013.

[67] Lynette Johnson-Read, Anthony Chmiel, Emery Schubert, and Joe Wolfe. Performing lieder: expert perspectives and comparison of vibrato and singer's formant with opera singers. *Journal of Voice*, 29(5), 2015.

[68] Maurílio N Vieira, José Eduardo de C Silva, and Hani C Yehia. Vibrato and tremor extent spectrum: Algorithm and applications. *The Journal of the Acoustical Society of America*, 130(1):EL1–EL7, 2011.

[69] Mingfeng Zhang, Mark Bocko, and James Beauchamp. Measurement and analysis of musical vibrato parameters. In *Proceedings of Meetings on Acoustics 169ASA*, volume 23, page 035004. Acoustical Society of America, 2015.

[70] Hee-Suk Pang, Jun-seok Lim, and Seokjin Lee. Discrete fourier transform-based method for analysis of a vibrato tone. *Journal of New Music Research*, 49(4):307–319, 2020.

[71] Petr Janata and Scott T Grafton. Swinging in the brain: shared neural substrates for behaviors related to sequencing and music. *Nature neuroscience*, 6(7):682–687, 0..3.

[72] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer music journal*, 29(1):34–54, 2005.

[73] Patrick Gomez and Brigitta Danuser. Relationships between musical structure and psychophysiological measures of emotion. *Emotion*, 7(2):377, 2007.

[74] L Henry Shaffer. How to interpret music. 1992.

[75] Simon Dixon. Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects*, volume 120, pages 133–137. Citeseer, 2006.

[76] Sebastian Böck, Andreas Arzt, Florian Krebs, and Markus Schedl. Online real-time onset detection with recurrent neural networks. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12), York, UK*. sn, 2012.

[77] 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.

[78] 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*. sn, 2013.

[79] Jedrzej Mońko and Bartłomiej Stasiak. Note onset detection with a convolutional neural network in recordings of bowed string instruments. In *International Conference on Multimedia Communications, Services and Security*, pages 173–185. Springer, 2017.

[80] Sebastian Böck and Markus Schedl. Enhanced beat tracking with context-aware neural networks. In *Proc. Int. Conf. Digital Audio Effects*, pages 135–139, 2011.

[81] EP MatthewDavies and Sebastian Böck. Temporal convolutional networks for musical audio beat tracking. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[82] Mojtaba Heydari and Zhiyao Duan. Don't look back: An online beat tracking method using rnn and enhanced particle filtering. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 236–240. IEEE, 2021.

[83] Matteo Lionello and Rafael Ramirez. A machine learning approach to violin vibrato modelling in audio performances and a didactic applica-

tion for mobile devices. In *Proceedings of the 15th Sound and Music Computing Conference (SMC)*, pages 347–353, 2018.

[84] Reinier Plomp. Timbre as a multi-dimensional attribute of complex tones. *Frequency analysis and periodicity detection in hearing*, pages 397–414, 1970.

[85] Anne Caclin, Stephen McAdams, Bennett K Smith, and Suzanne Winsberg. Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones. *The Journal of the Acoustical Society of America*, 118(1):471–482, 2005.

[86] Knut Guettler and Anders Askenfelt. Acceptance limits for the duration of pre-helmholtz transients in bowed string attacks. *The Journal of the Acoustical Society of America*, 101(5):2903–2913, 1997.

[87] JA Charles, Derry Fitzgerald, and E Coyleo. Violin timbre space features. In *Irish Signals and Systems Conference*, pages 471–476. Citeseer, 2006.

[88] Erwin Schoonderwaldt. The violinist's sound palette: spectral centroid, pitch flattening and anomalous low frequencies. *Acta Acustica united with acustica*, 95(5):901–914, 2009.

[89] Alfonso Perez, Jordi Bonada, Esteban Maestre, Enric Guaus, and Merlijn Blaauw. Score level timbre transformations of violin sounds. In *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08), Espoo, Finland*, 2008.

[90] Savvas Kazazis, Nicholas Esterer, Philippe Depalle, and Stephen McAdams. A performance evaluation of the timbre toolbox and the mirtoolbox on calibrated test sounds. In *Proceedings of the 2017 International Symposium on Musical Acoustics*, pages 144–147, 2017.

[91] Emery Schubert, Joe Wolfe, Alex Tarnopolsky, et al. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proceedings of the international conference on music perception and cognition, North Western University, Illinois*, pages 112–116. sn, 2004.

[92] Theodoros Giannakopoulos and Aggelos Pikrakis. *Introduction to audio analysis: a MATLAB® approach*. Academic Press, 2014.

[93] Jun Yang, Fa-Long Luo, and Arye Nehorai. Spectral contrast enhancement: Algorithms and comparisons. *Speech Communication*, 39(1-2):33–46, 2003.

[94] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116. IEEE, 2002.

[95] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis. Comparative evaluation of various mfcc implementations on the speaker verification task. In *Proceedings of the SPECOM*, volume 1, pages 191–194, 2005.

[96] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[97] Shin-Cheol Lim, Jong-Seol Lee, Sei-Jin Jang, Soek-Pil Lee, and Moo Young Kim. Music-genre classification system based on spectro-temporal features and feature selection. *IEEE Transactions on Consumer Electronics*, 58(4):1262–1268, 2012.

[98] TR Jayanthi Kumari and HS Jayanna. Comparison of lpcc and mfcc features and gmm and gmm-ubm modeling for limited data speaker verification. In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–6. IEEE, 2014.

[99] Tushar Ratanpara and Narendra Patel. Singer identification using mfcc and lpc coefficients from indian video songs. In *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1*, pages 275–282. Springer, 2015.

[100] DS Shete, SB Patil, and S Patil. Zero crossing rate and energy of the speech signal of devanagari script. *IOSR-JVSP*, 4(1):1–5, 2014.

[101] Arijit Ghosal, Rudrasis Chakraborty, Ractim Chakraborty, Swagata Haty, Bibhas Chandra Dhara, and Sanjoy Kumar Saha. Speech/music classification using occurrence pattern of zcr and ste. In *2009 Third International Symposium on Intelligent Information Technology Application*, volume 3, pages 435–438. IEEE, 2009.

[102] Sumit Kumar Banchhor and Arif Khan. Musical instrument recognition using zero crossing rate and short-time energy. *Musical Instrument*, 1(3):1–4, 2012.

[103] Fabien Gouyon, François Pachet, Olivier Delerue, et al. On the use of zero-crossing rate for an application of classification of percussive sounds. In *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*, volume 5, page 16. Citeseer, 2000.

[104] David Roxbee Cox. *Principles of statistical inference*. Cambridge university press, 2006.

[105] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[106] Dennis Howitt and Duncan Cramer. *Introduction to statistics in psychology*. Pearson education, 2007.

[107] Charles Stangor. *Research methods for the behavioral sciences*. Cengage Learning, 2014.

[108] Ju-Chiang Wang, Hsin-Min Wang, and Gert Lanckriet. A histogram density modeling approach to music emotion recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 698–702. IEEE, 2015.

[109] Athanasios Lykartsis, Chih-Wei Wu, and Alexander Lerch. Beat histogram features from nmf-based novelty functions for music classification. *10.14279/depositonce-9530*, 2015.

[110] Michael I Mandel and Daniel PW Ellis. Song-level features and support vector machines for music classification. 2005.

[111] Prashant Lahane and Arun Kumar Sangaiah. An approach to eeg based emotion recognition and classification using kernel density estimation. *Procedia Computer Science*, 48:574–581, 2015.

[112] Yi-Hsuan Yang and Homer H Chen. Prediction of the distribution of perceived music emotions using discrete samples. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2184–2196, 2011.

[113] Rajeev Rajan and Hema A Murthy. Music genre classification by fusion of modified group delay and melodic features. In *2017 Twenty-third National Conference on Communications (NCC)*, pages 1–6. IEEE, 2017.

[114] Douglas A Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech communication*, 17(1-2):91–108, 1995.

[115] Lukas Burget, Pavel Matejka, Petr Schwarz, Ondrej Glembek, and Jan Honza Cernocky. Analysis of feature extraction and channel com-

pensation in a gmm speaker recognition system. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):1979–1986, 2007.

[116] William M Campbell, Douglas E Sturim, and Douglas A Reynolds. Support vector machines using gmm supervectors for speaker verification. *IEEE signal processing letters*, 13(5):308–311, 2006.

[117] Christophe Charbuillet, Damien Tardieu, Geoffroy Peeters, et al. Gmm supervector for content based music similarity. In *International Conference on Digital Audio Effects, Paris, France*, pages 425–428, 2011.

[118] Ji-Hyun Song, Kye-Hwan Lee, Joon-Hyuk Chang, Jong Kyu Kim, and Nam Soo Kim. Analysis and improvement of speech/music classification for 3gpp2 smv based on gmm. *IEEE Signal Processing Letters*, 15:103–106, 2008.

[119] Roshni Ajayakumar and Rajeev Rajan. Predominant instrument recognition in polyphonic music using gmm-dnn framework. In *2020 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2020.

[120] Chih-Wen Weng, Cheng-Yuan Lin, and Jyh-Shing Roger Jang. Music instrument identification using mfcc: Erhu as an example. In *Proc. 9th Int. Conf. of the Asia Pacific Society for Ethnomusicology (Phnom Penh, Cambodia, 2004)*, pages 42–43. Citeseer, 2004.

[121] Arthur Flexer, Dominik Schnitzer, Martin Gasser, and Gerhard Widmer. Playlist generation using start and end songs. In *ISMIR*, volume 8, pages 173–178, 2008.

[122] Brian McFee, Luke Barrington, and Gert Lanckriet. Learning content similarity for music recommendation. *IEEE transactions on audio, speech, and language processing*, 20(8):2207–2218, 2012.

[123] Mathieu Barthet, György Fazekas, and Mark Sandler. Music emotion recognition: From content-to context-based models. In *International Symposium on Computer Music Modeling and Retrieval*, pages 228–252. Springer, 2012.

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

[125] 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, 2010.

[126] Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2-3):89–119, 2012.

[127] 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.

[128] Jean-Julien Aucouturier, Francois Pachet, et al. Music similarity measures: What's the use? In *ISMIR*, pages 13–17, 2002.

[129] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[130] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE, 2004.

[131] Jesper Højvang Jensen, Mads Græsbøll Christensen, Daniel PW Ellis, and Søren Holdt Jensen. Quantitative analysis of a common audio sim-

ilarity measure. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):693–703, 2009.

[132] Jacob Goldberger, Shiri Gordon, Hayit Greenspan, et al. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *ICCV*, volume 3, pages 487–493, 2003.

[133] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[134] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[135] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.

[136] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.

[137] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[138] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[139] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[140] Wei Chun Chiang, Jeen Shing Wang, and Yu Liang Hsu. A music emotion recognition algorithm with hierarchical svm based classifiers. In

2014 International Symposium on Computer, Consumer and Control, pages 1249–1252. IEEE, 2014.

[141] Gursimran Kour and Neha Mehan. Music genre classification using mfcc, svm and bpnn. *International Journal of Computer Applications*, 112(6), 2015.

[142] Jing Liu and Lingyun Xie. Svm-based automatic classification of musical instruments. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 3, pages 669–673. IEEE, 2010.

[143] Craig Saunders, David R Hardoon, John Shawe-Taylor, and Gerhard Widmer. Using string kernels to identify famous performers from their playing style. *Intelligent Data Analysis*, 12(4):425–440, 2008.

[144] Rafael Ramirez, Esteban Maestre, and Xavier Serra. Automatic performer identification in commercial monophonic jazz performances. *Pattern recognition letters*, 31(12):1514–1523, 2010.

[145] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017.

[146] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. Deep learning techniques for music generation–a survey. *arXiv:1709.01620*, 2017.

[147] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2018.

[148] Maria V Valueva, NN Nagornov, Pavel A Lyakhov, Georgii V Valuev, and Nikolay I Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177:232–243, 2020.

[149] Andrzej Skowroński and Kunio Yamagata. *Frobenius algebras*, volume 12. European Mathematical Society, 2011.

[150] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[151] Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

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

[153] Monika Dörfler, Roswitha Bammer, and Thomas Grill. Inside the spectrogram: Convolutional neural networks in audio processing. In *2017 international conference on sampling theory and applications (SampTA)*, pages 152–155. IEEE, 2017.

[154] Michael Stein, Benjamin M Schubert, Matthias Gruhne, Gabriel Gatzsche, and Markus Mehnert. Evaluation and comparison of audio chroma feature extraction methods. In *Audio Engineering Society Convention 126*. Audio Engineering Society, 2009.

[155] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. A comparison of audio signal preprocessing methods for deep neural

networks on music tagging. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1870–1874. IEEE, 2018.

[156] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 6979–6983. IEEE, 2014.

[157] Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining deep convolutional neural networks on music classification. *arXiv preprint arXiv:1607.02444*, 2016.

[158] Michael Taenzer, Jakob Abeßer, Stylianos I Mimilakis, Christof Weiß, Meinard Müller, Hanna Lukashevich, and IDMT Fraunhofer. Investigating cnn-based instrument family recognition for western classical music recordings. In *ISMIR*, pages 612–619, 2019.

[159] Vincent Lostanlen and Carmine-Emanuele Cella. Deep convolutional networks on the pitch spiral for musical instrument recognition. *arXiv preprint arXiv:1605.06644*, 2016.

[160] Rachel M Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello. Deep salience representations for f0 estimation in polyphonic music. In *ISMIR*, pages 63–70, 2017.

[161] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[162] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system

for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008.

[163] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

[164] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[165] Quoc V Le et al. A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, 20:1–20, 2015.

[166] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015.

[167] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 18–26, 2015.

[168] Minz Won, Sanghyuk Chun, and Xavier Serra. Toward interpretable music tagging with self-attention. *arXiv preprint arXiv:1906.04972*, 2019.

[169] Stevo Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020.

[170] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[171] Ankit Narendrakumar Soni. Application and analysis of transfer learning-survey. *International Journal of Scientific Research and Engineering Development*, 1(2):272–278, 2018.

[172] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[173] Haoran Chen, Yaowei Wang, Yemin Shi, Ke Yan, Mengyue Geng, Yonghong Tian, and Tao Xiang. Deep transfer learning for person re-identification. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–5. IEEE, 2018.

[174] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[175] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim. Auralisation of deep convolutional neural networks: Listening to learned features. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, pages 26–30, 2015.

[176] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho.

Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

[177] Juri Opitz and Sebastian Burst. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*, 2019.

[178] S Madeh Piryonesi and Tamer E El-Diraby. Data analytics in asset management: Cost-effective prediction of the pavement condition index. *Journal of Infrastructure Systems*, 26(1):04019036, 2020.

[179] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[180] JJ Bosch, F Fuhrmann, and P Herrera. Irmas: a dataset for instrument recognition in musical audio signals, 2014.

[181] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 21(2):522–535, 2018.

[182] Luwei Yang, Khalid Z Rajab, and Elaine Chew. The filter diagonalisation method for music signal analysis: frame-wise vibrato detection and estimation. *Journal of Mathematics and Music*, 11(1):42–60, 2017.

[183] Henrik Von Coler and Axel Roebel. Vibrato detection using cross correlation between temporal energy and fundamental frequency. In *Audio Engineering Society Convention 131*. Audio Engineering Society, 2011.

[184] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1467–1468. ACM, 2010.

[185] Simon Dixon and Gerhard Widmer. Match: A music alignment tool chest. In *ISMIR*, pages 492–497, 2005.

[186] Erik Marchi, Giacomo Ferroni, Florian Eyben, Leonardo Gabrielli, Stefano Squartini, and Björn Schuller. Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2164–2168. IEEE, 2014.

[187] Qingchen Zhang, Laurence T Yang, Zhikui Chen, and Peng Li. A survey on deep learning for big data. *Information Fusion*, 42:146–157, 2018.

[188] Claudia Fritz, George Stoppani, Igartua Unai, Roberto Jardón Rico, Arroitajauregi Ander, and Luis Artola. The bilbao project: How violin makers match backs and tops to produce particular sorts of violins. In *International Symposium on Musical Acoustics*, 2019.

[189] Percy Goetschius. *Lessons in music form: A manual of analysis of all the structural factors and designs employed in musical composition.* Oliver Ditson Company, 1904.

[190] Anders Askenfelt. Measurement of the bowing parameters in violin playing. ii: Bow–bridge distance, dynamic range, and limits of bow force. *The Journal of the Acoustical Society of America*, 86(2):503–516, 1989.

[191] Laurel S Pardue, Christopher Harte, and Andrew P McPherson. A low-cost real-time tracking system for violin. *Journal of New Music Research*, 44(4):305–323, 2015.

[192] Esteban Maestre, Panagiotis Papiotis, Marco Marchini, Quim Llimona, Oscar Mayor, Alfonso Pérez, and Marcelo M Wanderley. Enriched

multimodal representations of music performances: Online access and visualization. *Ieee Multimedia*, 24(1):24–34, 2017.

[193] Knut Guettler. On the creation of the Helmholtz motion in bowed strings. *Acta Acustica united with Acustica*, 88(6):970–985, 2002.

[194] Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.

[195] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José R Zapata, and Xavier Serra. Essentia: an open-source library for sound and music analysis. 2013.

[196] R EBU. 128, loudness normalisation and permitted maximum level of audio signals. *EBU Recommendatiion, Geneva*, 2014.

[197] Alfonso Perez-Carrillo. Violin timbre navigator: Real-time visual feedback of violin bowing based on audio analysis and machine learning. In Ioannis Kompatsiaris, Benoit Huet, Vasileios Mezaris, Cathal Gurrin, Wen-Huang Cheng, and Stefanos Vrochidis, editors, *MultiMedia Modeling*, pages 182–193, Cham, 2019. Springer International Publishing.

[198] Anssi Klapuri and Manuel Davy. Signal processing methods for music transcription. 2007.

[199] Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora. *MPEG-7 audio and beyond: Audio content indexing and retrieval*. John Wiley & Sons, 2006.

[200] Olivier Lartillot and Petri Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International conference on digital audio effects*, volume 237, page 244. Bordeaux, 2007.

[201] Kristoffer Jensen. *Timbre models of musical sounds*. PhD thesis, Department of Computer Science, University of Copenhagen Copenhagen, 1999.

[202] Reinier Plomp and Willem Johannes Maria Levelt. Tonal consonance and critical bandwidth. *The journal of the Acoustical Society of America*, 38(4):548–560, 1965.

[203] Patrik N Juslin. Cue utilization in communication of emotion in music performance: Relating performance to perception. *Journal of Experimental Psychology: Human perception and performance*, 26(6):1797, 2000.

[204] Y Zhao, C Wang, G Fazekas, E Benetos, M Sandler, et al. Violinist identification based on vibrato features. EURASIP, 2021.

[205] Francesco Setragno, Massimiliano Zanoni, Fabio Antonacci, and Augusto Sarti. Feature-based timbral characterization of historical and modern violins. In *International Symposium on Musical Acoustics*, pages 90–93, 2017.

[206] Massimiliano Zanoni, Francesco Setragno, Fabio Antonacci, Augusto Sarti, György Fazekas, and Mark B Sandler. Training-based semantic descriptors modeling for violin quality sound characterization. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[207] Juhan Nam, Keunwoo Choi, Jongpil Lee, Szu-Yu Chou, and Yi-Hsuan Yang. Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach. *IEEE signal processing magazine*, 36(1):41–51, 2018.

[208] Tong Liu, Li Han, Liangkai Ma, and Dongwei Guo. Audio-based deep

music emotion recognition. In *AIP Conference Proceedings*, volume 1967, page 040021. AIP Publishing LLC, 2018.

[209] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. *Deep learning techniques for music generation*, volume 1. Springer, 2020.

[210] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, pages 387–392. Citeseer, 2009.

[211] Thierry Bertin-Mahieux, Daniel Ellis, Brian Whitman, and Paul Lamere. The million song dataset. 2011.

[212] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. 2019.

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

[214] Ron Kohavi and Dan Sommerfield. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *KDD*, pages 192–197, 1995.

[215] Andreas Bugler, Bryan Pardo, and Prem Seetharaman. A study of transfer learning in music source separation. *arXiv preprint arXiv:2010.12650*, 2020.

[216] Deepanway Ghosal and Maheshkumar H Kolekar. Music genre recognition using deep neural networks and transfer learning. In *Interspeech*, pages 2087–2091, 2018.

[217] Beici Liang and Minwei Gu. Music genre classification using transfer learning. In *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 392–393. IEEE, 2020.

[218] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

[219] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. Look, listen, and learn more: Design choices for deep audio embeddings. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856. IEEE, 2019.

[220] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models. *arXiv arXiv:2006.00751*, 2020.

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

[222] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 366–370. IEEE, 2018.

[223] Jordi Pons and Xavier Serra. musicnn: Pre-trained convolutional neural networks for music audio tagging. *arXiv preprint arXiv:1909.06654*, 2019.

[224] Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serrc. Data-driven harmonic filters for audio representation learning. In *International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 536–540. IEEE, 2020.

[225] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[226] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[227] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

[228] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[229] Rajesh Sangeetha and NJ Nalini. Singer identification using mfcc and crp features with support vector machines. In *Computational Intelligence in Pattern Recognition*, pages 295–306. Springer, 2020.