ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MIXPREP: MACHINE LEARNING-BASED MULTITRACK MIX PREPARATION ASSISTANT

Ph.D. THESIS

İsmet Emre YÜCEL

Department of Music

Music Programme

AUGUST 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MIXPREP: MACHINE LEARNING-BASED MULTITRACK MIX PREPARATION ASSISTANT

Ph.D. THESIS

İsmet Emre YÜCEL
(409152003)

Department of Music

Music Programme

Thesis Advisor: Doç. Dr. Taylan ÖZDEMİR

AUGUST 2022

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MIXPREP: ÇOK KANALLI SES MİKSAJ HAZIRLIĞI İÇİN MAKİNE ÖĞRENMESİ TABANLI ASİSTAN

DOKTORA TEZİ

İsmet Emre YÜCEL
(409152003)

Müzik Ana Bilim Dalı

Müzik Doktora Programı

Tez Danışmanı: Doç. Dr. Taylan ÖZDEMİR

AĞUSTOS 2022

İsmet Emre YÜCEL, a Ph.D. student of ITU Graduate School student ID 409152003, successfully defended the thesis/dissertation entitled "MIXPREP: MACHINE LEARNING-BASED MULTITRACK MIX PREPARATION ASSISTANT", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Taylan ÖZDEMİR** İstanbul Technical University	.....
Jury Members :	**Prof. Dr. Can KARADOĞAN** Istanbul Technical University	.....
	Prof. Dr. Arda EDEN Yıldız Technical University	.....
	Asst. Prof. Dr. Ali BOYACI İstanbul Ticaret University	.....
	Asst. Prof. Dr. Ozan SARIER Istanbul Technical University	.....

Date of Submission : 23 June 2022
Date of Defense : 12 August 2022

Yaşam boyu destekleri için annem ve babama,

Sabır ve desteğini her zaman hissettiğim sevgili eşim Meral'e,

ve biricik kızım Ela Bahar'a...

FOREWORD

First of all, I would like to express my gratitude to the administration and the faculty members of MIAM for providing an exclusive academic environment. During my PhD studies, I have had the opportunity to work with such valuable lecturers and met very talented students. It has been a great honor to study sound engineering with Pieter SNAPPER, Can KARADOĞAN, Taylan ÖZDEMİR, and sonic arts with Reuben de LAUTOUR.

Secondly, I am also very grateful to the administration and academic personnel of Sakarya University State Conservatory for their support alongside their efforts in understanding my research.

I also want to thank my advisor Taylan ÖZDEMİR for his precious support; Arda EDEN, who promoted this study with some eye-opening ideas; and Can KARADOĞAN, who has an important role as a committee member in addition to his guidance.

Lastly, I owe my wife a debt of gratitude, without whose support, patience, and encouragement this study could not have been successfully completed.

August 2022

İsmet Emre YÜCEL
(Research Assistant)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ACA : Audio Content Anaylsis
AI : Artificial Intelligence
ANN : Artificial Neural Network
API : Application Programming Interface
CD : Compact Disc
CENS : Chroma energy normalized statistics
CNN : Convolutional Neural Network
DCN : Deep Convolutional Network
DAW : Digital Audio Workstation
DC : Direct Current
DFT : Discrete Fourier Transform
DSP : Digital Signal Processing
DTW : Dynamic Time Warping
FDR : False Discovery Rates
FFT : Fast Fourier Transform
FNR : False Negative Rates
GA : Genetic Algorithm
GMM : Gaussian Mixture Model
GUI : Graphical User Interface
HMM : Hidden Markov Model
ICA : Independent Component Analysis
IDAW : Intelligent Digital Audio Workstation
IMP : Intelligent Music Production
IRMAS : Instrument Recognition in Musical Audio Signals
KDD : Knowledge Discovery in Databases
KNN : K-Nearast Neigbour
LDA : Linear Discriminant Analysis
LQV : Learning Vector Quantization
MDF : Multiscale Fractal Dimension
MIDI : Musical Instrument Digital Interface
MFCC : Mel-Frequency Cepstral Coefficients
MIR : Music Information Retrieval
MISDIN : Musical Instrument Sample Database of Isolated Notes (database)
ML : Machine Learning
MLP : Multilayer Perceptron
MPEG : Moving Pictures Expert Groups
MUMS : MacGill University Master Samples
NNC : Nearest Neighbour Classification
OverCs : Combine a compact description of MFCC
PPV : Positive Predictive Values
RBF : Radial Basis Function
RMS : Root Mean Square
RNN : Recurrent Neural Network
RWC : Real World Computing (audio database)

SBS : Sequential Backward Selection
SNR : Signal to Noise Ratio
SOMNN : Self-organizing Mapping Neural Network
SparCs : Sparse Representation of MFCC
SSL : Semi-supervised Learning
STFT : Short Term Fourier Transform
SVM : Support Vector Machine
TMIR : Transient length instrument recognition
TPR : True Positive Rates
UIOWA : University of Iowa Musical Instrument Samples
VCA : Voltage Controlled Amplifier
VST : Virtual Studio Technology
ZCR : Zero crossing rate

SYMBOLS

T	: Period
f	: Frequency
m	: Mass
θ	: Phase
t	: Time
A	: Amplitude
fn	**:** n'th harmonic
MxN	**:** Numbers of Rows and Columns
p	: Probability

LIST OF TABLES

Page

LIST OF FIGURES

Page

MIXPREP: MACHINE LEARNING-BASED MULTITRACK MIX PREPARATION ASSISTANT

SUMMARY

Music production is a general term for describing a set of complicated processes where artistic and technical efforts are involved. Besides the artistic part, the technical side of some parts has regular iterative works. This study focuses on the mix preparation step of the multitrack audio mixing stage in music production by seeking an automatic software solution regarding the intelligent music production paradigm.

The structure of the dissertation consists of four components: Theoretical background with fundamental definitions of knowledge both in music production analysis, instrument recognition theories and applications, the approach and explanation of the development of the proposed assistant software, and last but not least, an experiment stage comprising of performance testing with many multitrack projects.

Before diving into the development stage, the perspective and the definition of the mix preparation are presented after introducing the music production with a brief historical background. Afterwards, delineation of the intelligent music production research field apart from subjective interests takes part.

Instrument recognition literature takes an important part in the conceptualization of the automatic mix preparation solution. Because of that, an extensive historical background in the instrument recognition field is given without getting into redundant theoretical aspects. Apart from that, a reasonable amount of information about the definition of the fundamental concepts of digital audio, audio content analysis and machine learning seemed appropriate to be mentioned since the audience of this research addressed in the music technology field.

After providing the fundamental theoretical background, the software development approach for the mix preparation assistant is presented. This section explains the software structure by stating the basic requirements of the mix preparation regarding design concerns of the graphical user interface (GUI) consideration for practical usage. The main issues are the GUI layout, software usage, and building a dataset with a related machine learning model.

Eventually, a loop-based audio dataset creation approach and ML model are put forward by testing their performance with many audio files from 80 multitrack audio projects in four musical genres (Pop, Rock, Jazz, Electronic/Dance). The experiment is set concerning instrument families provided in the dataset and genre-related performance estimations of each one. The results were interpreted by accentuating the crucial points of implementing the ML-based mix preparation solution. Detailed evaluation results are in the appendices.

This study proposes a concept of intelligent mix preparation software by providing a methodology for the design concept and application.

MIXPREP: ÇOK KANALLI SES MİKSAJ HAZIRLIĞI İÇİN MAKİNE ÖĞRENMESİ TABANLI ASİSTAN

ÖZET

Müzik prodüksiyonu teknik ve artistik birçok karmaşık işlemin dahil olduğu süreçler bütününü temsil eden bir terim olarak karşımıza çıkmaktadır. Ses mühendisliği açısından bakıldığında prodüksiyon; kayıt, miksaj ve kalıplama (mastering) şeklinde üç ana aşamada ele alınır. Ses mühendisleri üretilecek müziğin tarzını, tüketim ihtiyaçlarını, geleneksel uygulamaların sağladığı birikim ile; ancak güncel ihtiyaçları da göz önünde bulundurarak, sanatsal ve estetik açıdan değerlendirip prodüksiyon aşamalarını gerçekleştirirler. Söz konusu bu sanatsal ve estetik değerlendirmelerin kayıt mekânının akustiği, mikrofon tercihleri ve yerleşimleri, kullanılacak ses işlemcilerin tercihi, stereo alanda çalgıların yerleşimi, dinamik aralık kullanımı ve daha birçok teknik kararların verilmesinde önemli bir rolü vardır. Sanatsal ve estetik kararlar kısmı bir yana, müzik prodüksiyonunun bazı adımlarının teknik kısımları kendini tekrarlayan rutinlere sahiptir. Bu rutinler, prodüksiyonun kapsam ve büyüklüğüne bağlı olarak uzamakta ve ses mühendisinin üzerinde çalıştığı albüme estetik açıdan harcaması gereken enerjiyi ve zamanı çalmaktadır. Bu rutinlerden en dikkat çekici olanı çok kanallı ses miksaj hazırlığı uygulamasıdır. Miksaj hazırlığı, miks mühendislerinin çok kanallı miksaja başlamadan önce, DAW (Digital Audio Workstation) projelerini kendi iş akışlarına göre ayarladıkları, projeyi oluşturan kanallar üzerinde gerekli gördükleri düzenleme ve düzeltmeleri yaptıkları tüm uygulamaları kapsamaktadır. Miksaj hazırlık aşaması, her ne kadar ses mühendisleri tarafından gerekli bir rutin olarak değerlendirilse de özellikle büyük albüm projelerinde her bir şarkının sahip olduğu onlarca kanal (audio tracks) düşünüldüğünde, oldukça zaman alan bir uygulamadır. Bu çalışmanın amacı, yeni bir araştırma alanı olarak karşımıza çıkan akıllı müzik prodüksiyonu (intelligent music production) çalışmaları kapsamında, müzik prodüksiyonundaki çok kanallı ses miksaj hazırlığını otomatikleştirmeyi hedefleyen bir yöntem geliştirmek ve bu yöntemi bir yazılım çözümü olarak ortaya koymaktır.

Bu çalışma altı bölümden oluşmaktadır. birinci bölümde; amaç, kapsam ve hedefler belirtilmiş, ikinci bölümde; müzik prodüksiyonunun teorik arka planı ve temel kavramlarına yer verilmiş, üçüncü bölümde; bu çalışmanın yöntemi bağlamında yer alan çalgı tanımlama çalışmalarının literatürü, ilgili teori ve uygulamaları açıklanmış, dördüncü bölümde; çok kanallı ses miksaj hazırlığı yazılımının tasarım süreci, bu süreçte takip edilen yöntem ve prensipler (tasarım yaklaşımı, yazılım arayüzü ve kullanımı vb.) ortaya konmuş, beşinci bölümde; geliştirilen yazılım ile çok kanallı proje dosyaları test edilerek, yazılıma entegre edilen çalgı tanımlama temelli makine öğrenmesi modelinin başarı sonuçları ölçülmüş ve değerlendirilmiştir. Altıncı ve son bölüm olan sonuç kısmında ise bu alanda yapılacak olan ileriki çalışmalar için araştırma önerileri sunulmuştur.

İkinci bölümde, müzik prodüksiyonunun tanımı ve tarihsel gelişiminin zaman içinde değişimi manyetik bant öncesi dönemden başlanarak açıklanmıştır. Özellikle analog

dönemin parladığı 60'lardan sonra, manyetik bant teknolojisinin bir kayıt ortamı olarak müzik endüstrisine ve dolayısıyla ses kayıt stüdyolarına girişi, buna bağlı olarak da zaman içerisinde oluşan teknik dil ve yıllar içerisinde şekillenen prodüksiyon süreçleri günümüz prodüksiyon rutinlerini anlamak açısından önemli bir bilgi birikimi sunmaktadır. Bu birikim, her ne kadar günümüz müzik prodüksiyonu uygulamaları artık tamamen sayısal kayıt sistemler ile gerçekleştirilse de geçerliliğini korumakta ve halen faydalanılması açısından önemlidir. Yine bu kısımda, günümüzdeki prodüksiyon aşamaları belirtilmiş ve ses mühendisliği kavramı bu bağlamda açıklanmıştır. Müzik prodüksiyonunun bir alt aşaması olarak çok kanallı miksaj hazırlığı detaylı bir şekilde açıklanmış ve önemi ortaya konmuştur. Bu bağlamda miksaj hazırlığı; kanalların proje içerisindeki organizasyonları (track organization), miks mühendisinin müzik tarzına bağlı öznel tercihleri (style dependent) ve ses kanallarının içerikleri üzerinde yapılacak düzenleme/düzeltmeler (edit-based) olacak şekilde kategorize edilmiştir. Kanalların organizasyonu; alt gruplama, alt grup renklendirme ve isimlendirmeleri, bu alt gruplar içerisindeki sinyal yönlendirmelerinin yapıldığı genel uygulamaları kapsamaktadır. Müzik tarzına bağlı öznel tercihler ise, bir miks mühendisinin, yapılacak alt gruplamalar ve bu alt gruplarındaki sinyal yollarının (buss) proje içerisindeki yerleşimleri, ses kanalları ve/veya sinyal yollarında kullanacak ses işlemcileri ve diğer ses efektleri hakkındaki tercihleridir. Kanal içerik düzeltmeleri temelli (edit-based) uygulamalar ise her bir kanalın tek tek dinlenmesini gerektiren; kesme/biçme, ad değiştirme, faz düzeltme perde düzeltme (pitch correction) gibi işlemlerin yapıldığı, kontrolleri ifade etmektedir. Ardından, bu tezin hazırlamasında çıkış noktası olan akıllı müzik prodüksiyonu çalışma alanı ve hedefleri kısaca açıklanmıştır. Akıllı müzik prodüksiyonu çalışma alanı, yukarıda bahsedilen rutinleri otomatikleştirmeyi hedeflemektedir. Otomatik çok kanallı miksaj, akıllı ses işlemciler, müzik uygulamalarında semantik sistemler, oyunlar için otomatik ses üretimi vb. çözümler akıllı müzik prodüksiyonu çalışma alanı ile ilişkilidir.

Çalgı tanımlama çalışmaları ile ilgili literatürün, bu çalışmada yer alan otomatik miksaj hazırlık yazılımının tasarlanmasında önemli bir rolü vardır. Bu nedenle, çalgı tanımlama literatürünün tarihsel süreci, fazla teorik detaya girilmeden bu kısımda ortaya konmuştur. Böylece, bu alanda çalışacak olan araştırmacılar için değerli bir literatür özeti oluşturulması hedeflenmiştir. Literatür özetinden sonra sesin oluşumu, sayısal ses, sinyal, sayısal ses sinyali ve sayısal ses işleme ile ilgili temel bilgiler açıklanmıştır. Yine bu kısımda, çalgı tanımlama algoritmalarında kullanılan öznitelik çıkarım (feature extraction) yöntemleri ve makine öğrenmesi ile ilgili bazı temel kavramlar yeterli görüldüğü derecede; bu çalışmanın hitâp ettiği araştırmacı kitlesi göz önünde bulundurularak bölüm üçe eklenmiştir.

Yukarıda bahsedilen bilgilendirmelerden sonra, dördüncü bölümde; bu araştırma için otomatik miksaj hazırlık yazılımının geliştirilmesi sırasında takip edilen yöntem ve yaklaşımlar ortaya konmuştur. Yazılımın hangi DAW ile çalışacağı, geliştirileceği platform, içerisine gömülecek olan makine öğrenmesi kütüphanesinin seçimi; çalgı ailelerine uygun veri seti ve bu veri seti ile makine öğrenmesi modelinin oluşturulması gibi konular irdelenmiştir. Bu yazılım geliştirilirken, grafik arayüzünün basit ve kolay kullanılabilir olması, hızlı tepki süresine sahip olması, özelleştirilebilme ve çalgı tanımlama temelli makine öğrenmesi modeli ile yapılmış alt gruplandırmaların kullanıcı tarafından hızlı bir şekilde değiştirilebilmesine olanak sağlaması gibi kriterler ön planda tutulmuştur. Yukarıda bahsedilen miksaj hazırlık yaklaşımları göz önünde bulundurulduğunda, bu çalışma sadece kanalların proje içerisindeki organizasyonları ile kısıtlanmıştır. Buna göre bu yazılım; proje içerisindeki alt grupların oluşturması,

isim ve renklendirmelerinin yapılması; öte yandan, bu alt gruplardaki sinyal akışlarının düzenlenmesi görevlerini bir makine öğrenmesi yöntem ve modeli yardımı ile otomatik yapacak şekilde geliştirilmiştir. Bu bağlamda, yazılım geliştirme platformu olarak Python ve DAW olarak da REAPER tercih edilmiştir. Kullanıcı, yazılımın grafik arayüzünü kullanarak proje dosyasının içeriğindeki ses dosyalarını yazılım içerisine dahil eder. Ardından projedeki ses dosyaları kullanıcı tarafından on adet (grup A'dan, ... grup I'ya) ön tanımlı alt grupta için, grup ismi ve renklerini değiştirilerek organize edilebilir. Yine bu işlem yazılıma gömülü, daha önceden amaca yönelik bir ses kütüphanesi ile yaratılmış bir veri setinin kullanıldığı makine öğrenmesi modeli ile otomatik olarak da yaptırılabilir. Otomatik alt gruplama için bu çalışmada bir tekrarlı ses (audio loops) kütüphanesi kullanılmış ve bu ses kütüphanesi altı çalgı ailesi (vurmalılar, baslar, gitarlar, tuşlular, yaylılar ve üflemeliler) oluşturacak şeklinde düzenlenmiştir. Düzenlenen dosyalar ile ses içerik analizi yöntemleri kullanılarak zaman ve tını bazlı öznitelikler çıkartıldı ve bir veri seti meydana getirildi. Devamında, SVM makine öğrenmesi algoritması ile bu veri seti ile eğitilerek bu tez kapsamında belirlenen çalgı ailelerinin tanımlanması amacıyla bir makine modeli oluşturuldu. Öznitelik çıkarımı ve sınıflandırma temelli makine öğrenmesi modeli için pyAudioAnalysis kütüphanesi tercih edilmiştir. Normalde konsol bazlı çalışan bu kütüphaneye detaylı bir grafik arayüz eklenerek bir modül haline getirilmiş ve miksaj hazırlık programı içerisine gömülmüştür. Böylece kullanıcılara, tercih ettikleri öznitelikler ve istatistik hesaplamalara göre veri seti ve modeller oluşturabilmeleri için imkân sağlanmıştır.

Beşinci bölümde, bu çalışma kapsamında üretilen altı çalgı aileli veri seti ve ilgili makine öğrenmesi modeli, dört müzik tarzında (Pop, Rock, Caz, Elektronik/Dans) toplamda 80 adet çok kanallı müzik projesi ile ayrı ayrı test edilmiştir. Proje bazlı bu testte başarı ölçütü olarak, her bir ses dosyasının (kanal) ismi ile o ses dosyasının mevcut makine modeli tarafından yerleştirildiği çalgı ailesine uyumluluğu kriter olarak alınmıştır. Örneğin, drums.wav isimli bir ses dosyası eğer çalgı tanımlama algoritması tarafından davul ve perküsyon sınıfına yerleştirildiyse bu sonuç başarılı sayılmıştır. Bu test ile toplamda 1428 adet ses dosyası mevcut veri seti ve makine öğrenmesi modeli ile denenmiş ve bunlardan 924 tanesinin doğru sınıflandırıldığı görülmüştür. Böylece uygulamada mevcut modelin genel başarı oranının 64.71% olduğu gözlemlenmiştir. Çalgı grupları için başarı oranları, Davul ve Perküsyonlar için 84.80%, Baslar için 67.72%, Yaylılar için 61.11%, Gitarlar için 53.66%, Üflemeliler için 25.00% ve Tuşlular için ise 16.44% olduğu görülmüştür. Müzik tarzları açısından başarı oranları ise Rock için 68.52%, Pop için 65.89%, Caz için 67.20%, Elektronik/Dans için 57.10% olarak belirlenmiştir. Detaylı sonuçlara tezin ekler kısmında yer verilmiştir.

1. INTRODUCTION

Music production is a general term that contains the embodiment of artistic and technical processes which exist along with different layers of complexities from music composition to distribution. Previously, music production had required dedicated facilities which a limited number of people could reach until digital audio recording technologies became more accessible thanks to the contribution of personal computers. Now, the DAWs are essential tools for music production. Due to practical reasons and low expenses, the overwhelming majority of music creations appear in the DAWs as "in the box" at home studios or project studios with more affordable audio equipment.

Generally, music production consists of three steps: recording, mixing, and mastering. These steps conjugate into one mission wherein many sound sources are coherently assembled to support the musical narrativity containing different procedural difficulties and artistic choices of production. Whereas the steps benefit from common sound processing principles regarding the technical side of the production, each has its priority while dealing with a specific task. Before starting multitrack mixing, mix preparation is an essential routine that engineers frequently perform, particularly for larger projects. The mix preparation involves some procedures that importing tracks into a DAW, naming individual tracks correctly, putting the tracks in order according to their importance and musical functions, assigning colors to grouped tracks, creating channel groups, VCA and folder tracks, aux channels for the usage of subgroups and audio effects such as reverbs, delays, or parallel compression, track editing and so on.

Digital audio technologies broadly affect communication, consumption and listening experiences of the society. Voice recognition, audio fingerprinting, music recommendations systems are some technologies that have proved their commercial potentials over the years. On the other hand, in music applications, technology has led to many ground-breaking innovations and inventions such as beat detection, autotuning, audio time alignment, hardware emulations, and remote collaboration. Most of them may be categorized as "assistant tools", which help the engineer to do his/her job more quickly and precisely. Those kinds of audio technologies have

become available thanks to the combination of different disciplines. One essential discipline is audio content analysis (ACA), which is considered a subordinate of the music information retrieval (MIR) field. After data acquisition is made from audio by ACA methods, the audio data is interpreted and classified by machine learning (ML) or artificial intelligence (AI) methods. Implementation of these methods into any part of the music production has created a new concept: “Intelligent Music Production”.

This study proposes a new approach to the mix-preparation of music production by referring to instrument classification solutions based on ACA and machine learning methods to estimate audio subgroups.

1.1 The Aim, Scope, and Objectives of the Study

Mix preparation is a well-known and common practice for audio subgrouping, coloring, ordering (channels), rough mixing, etc., before a multitrack mix session starts. However, there is nowadays no attempt to make this practice a more effortless process. The DAWs do not provide any dedicated solution to execute this process automatically and intelligently. If mix preparation becomes automatic by the DAW or external software, engineers can start the multitrack mixing step earlier without losing their energy for reviewing and organizing individual tracks. This study investigates how an intelligent mix preparation assistant can be created with the available software tools. The proposed intelligent mix preparation assistant model can automatically subgroup the related audio files by coloring and naming with predefined parameters.

For mix-preparation, neither qualitative approaches over subjective preferences nor new techniques and methods are available in this study; yet, only some track organization related part was taken into account considering available practices. Moreover, this study is limited only to the existing audio analysis and machine learning algorithms to deal with the track organization-based mix preparation solution.

The first objective was to determine the feasibility and applicability of fundamental practices of the track organization-based mix preparation steps. Then the software features were discussed.

The second objective was to obtain the software development platform to implement the track organization on the mix preparation. The decision over the software development platform depends on whether this software works with DAW or as a

standalone. Since importing and organizing the audio files in a session will be an elaborate task, an open-source DAW as a testbed will be beneficial. Additionally, a GUI is a obligatory feature of the software to interact with users to control session preferences.

The third objective was to find an audio library and appropriate automatic subgrouping algorithm to estimate instrument families of the multitrack project files to create a dataset for the experiment. Because dedicated instrument group/family names must be known before the audio subgrouping tasks, a supervised ML algorithm seemed adequate to classify temporal and spectral audio features.

1.2 Methodology

This study is practiced upon open-source audio analysis and ML libraries in the Python programming language, thus requires an audio dataset while distinguishing predefined instrument classes by a supervised ML. The low-level audio feature extraction and statistical calculations are essential data acquisition techniques that enable the creation of audio datasets from audio sources. However, the necessity of gathering and organizing many audio files on purpose to attain a dataset from various audio sources may become quite challenging. Therefore, since it has royalty-free content, the Apple Jam Packs loop library is preferred in this study. The library was revised, organized, and categorized into individual folders regarding instrument classes as Drums&percussion, Bass, Guitars, Keyboards, Strings, and Wind; then the dataset was procured with the process of feature extraction.

The audio feature extraction process comprises temporal and spectral features and a related dataset used for the ML model on SVM with RBF kernel. The dataset is also checked with all ML algorithms provided with MATLAB.

Lastly, the model is tested with 80 multitrack project files from an open internet source. The projects were collected considering four genres for which 20 projects were available. The model success rates were examined by comparing the annotations of each audio file and their instrument family estimation results by the ML model.

2. MUSIC PRODUCTION

Music production defines the music creation process that combines different technical and artistic skills. Hepworth-Sawyer and Russ (2010) describe music production as "…an expression of the creative and artistic development of music both in and out of the studio." (p 17). Previously, music production took place in dedicated environments where limited technologies and human sources were used. The traditional music production hierarchy comprised of producer, engineer, assistant engineer, and tape operators whose responsibilities were clearly specified during the music creation process (Hepworth-Sawyer & Golding, 2010).

Over the years, music production standards and habits have been changed by the emergence of new technologies. After the advent and requisition, tape recorders became one of the most crucial revolutions in recording technologies. Alexander (1994) highlighted that pretape technologies were expensive and unforgiving over mistakes in music production (p. 119). At that time, musicians or bands had to be carefully placed into a venue and recorded only with one microphone into vinyl, and there was no chance to correct recording mistakes except recording them again and again. In the 1960s the availability of 4-8 track tape recorders brought a new methodology which is called overdubbing by which the musicians could listen to previously recorded tracks while their performances were being recorded over a new track. Also, engineers could change individual channel levels thanks to multitrack tape recorders. In the 1970s, 24 tracks recording tapes became standard in the industry, so almost every instrument started to be recorded on a dedicated channel without concerning leakage; yet, clear and dry recording opportunities changed the design of studios' infrastructures by requiring isolated booths and reverberation chambers. In the 1980s, the track numbers increased up to 48-72 in tape recorders, which led to further separation over the recordings, causing promotion of the expectation to create "bigger sound" with newly incoming digital reverberation algorithms. This expectation may explain why the excessive usage of reverberation and gating on audio tracks (especially in drum recordings) was prominent. In the 1990s, 24-32-48 tracked

digital recording systems were adapted to the studio environments. The rising of digital technology had also changed music production using loop and audio replacements, and computers started to take place in the studio environments. After the millennium, digital technologies were approved providing some advantages, namely distribution and migration, low budgets, infinite number of tracks, easy session recall and access; however, this induced the questioning necessity of large recording studios due to their affordability and caused the proliferation of small project studios (aka home studios) where recording and mixing are conducted with low budget (Weekhout, 2019, p. 7–11).

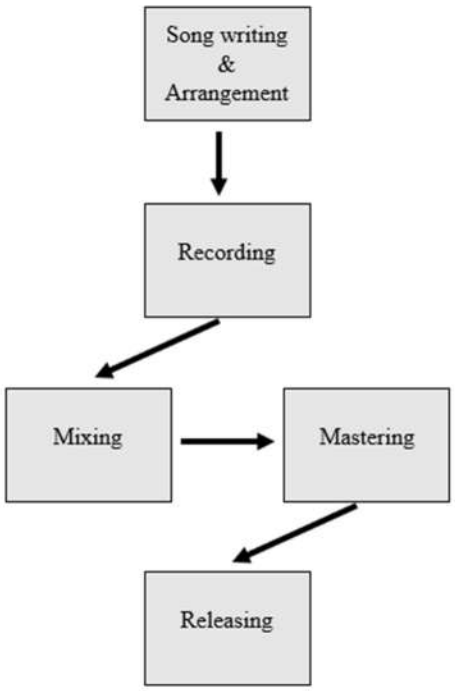

Figure 2.1: The music production steps.

In the general sense, the steps of music production consist of song writing, arrangement, recording, mixing, mastering, and releasing routines (Figure 2.1). Even though the term "sound engineering" is associated with a profession in which the technical skills and abilities seem to dominate, musical and artistic skills are also required. Furthermore, sound engineering may also entail a certain level of knowledge and artistic skills to carry on their musical tendencies in low-budget productions, which is also quite common in today's music.

For mixing, in the early days of the 1950s, there were one channel mono tracks, and recordings required minimal mixing. The innovation of Self-Sync technology (in 1955) allowed overdubbing, and an increasing number of tracks on multichannel tape

recorders in studios changed mixing philosophy. Today's mixing approaches are very similar after 24 track tape recorders showed up in 1975. While the number of channels increased, mixing turned into complex tasks needing extra labor at that time. The complexity was solved by the arrival of automation, total recall, and reset functions into the mixing consoles. After 2001, the "mixing in the box" term showed up because music production became a computer-centralized process, and budget drop in album projects urged mix engineers to work from their home studios (Owsinski, 2014b, p. 4).

Even though the concept of mastering has changed over the years, the principle behind it stays the same as making the album sounds sonically equal for professional and home systems. Previously, mastering was considered to make songs of an album sonically similar and coherent; but today, it is a process of fine-tuning the level, frequency, and metadata of a track for release. We observe that mastering appeared as a process by the invention and commercial usage of magnetic tapes in the recording industry. In the past, the mastering engineers, called transfer engineers because their job was to transfer tape records of albums to vinyl masters for releasing them. That was a painstaking task due to level changes possibly damaging the master vinyl (even cutting stylus or lathe) or causing a low SNR signal on the disc. The distinction between job descriptions of recording and mastering engineers became apparent with the innovation of overdubbing technology in 1955. Afterwards, the availability of commercial stereo vinyl pushed the industry to another level (in 1957). At that time, mastering engineers were also called cutters. They started to use equalization and compression to make discs louder because louder songs attracted the listener's attention on the radio, resulting in increasing album sales. Since the CDs came as a new medium in 1982, mastering was the first music production step adapted to this technology. In 1995, the compressed audio formats (such as MP3s) dawned with size and distribution advantages, and mastering engineers had to learn how to get the best results for themselves. After 2002, the in-the-box workflow was an inevitable approach to mastering like mixing (Owsinski, 2014a, pp. 4–7).

Nevertheless, even the steps of production stayed more or less same, its boundaries have evolved after the emergence of the digital recording systems by pervading usage of DAWs from semi-professionals to novices over the years. The affordability of the recording technologies allows artists to create music at home without building a dedicated recording studio. This type of music production is beneficial for "bedroom

musicians" who want to make music with low budgets, whether professionally or not, and "bedroom production" contributes to the proliferation of music creation, too. Thanks to technology, today, artists have countless opportunities to create and share their music without bothering with the overwhelming technical rigors of the production that were required to be handled by certain efforts and skills regarding conventional music production steps in the past.

2.1 Definition of the Mix Preparation

It is a well-known fact that mix preparation is a rational habit where the session organization proceeds regarding some categorical perspectives of the building strategies of the mix by the engineers. The fundamental approach is to organize the project tracks according to their functions in an arrangement. The duration of mix preparation depends on the scope of the project. Owsinsky (2014b) explains the mix preparation in two categories: Session preparation and personal preparation. Owsinsky's categorization over session preparation can be summarized as follows:

- **Track organization:** Arrange and reorder tracks, give colors to tracks, create groups and subgroups, use section markers.
- **Style dependent:** It is related to how engineers prefer to set their mixing workflows while approaching a specific genre of music; what audio processing will happen during the mixing; what type of equalizers, compressors, limiters, reverbs and other audio plugins and their replacements will be applied to the tracks.
- **Edit based:** This category comprises some efforts that engineers do such as tweaking the timing of the tracks, eliminating noises, checking the tracks' labels and fades, making a session file copy, tuning audio tracks (especially for vocals), deactivating/hiding tracks or deleting empty tracks, and so on.

Mix preparation is very critical to achieve professional-level results in music production. Besides, in professional studios, it is easy to find any dedicated or volunteering staff who takes on the mix preparation task. But, in low budget projects or smaller studios (like home studios), this task is a burden for the mixing engineer (Senior, 2011).

2.2 Intelligent Music Production (IMP)

In the general description, music production is a process that combines creativity and artistic developments in music, whether it is held in a studio environment or out (Hepworth-Sawyer & Golding, 2010). When referring to the term "music production", we depict a complex structure that conjugates a chain of tasks such as composition, arrangement, recording, editing, mixing, and mastering.

Application of Intelligence Systems (IS) has been studied widely since the 1990s. The emergence of IS was to solve intricate and challenging problems that could not be solved by mathematically modelled traditional methods. Those systems can be designed to solve a problem by conduction soft computing techniques[1]or AI (C. et al., 2009, p. 19).

In May 2000, an outstanding article was published in the journal of the Audio Engineering Society. On that paper, a vision of the emergence of a new research field combining intelligent systems and music production was given as "intelligent assistants" (Moorer, 2000). This idea developed in time and became the concept that Intelligent Music Production (IMP). Today, a good amount of literature exists in that field.

According to De Man et al. (2019), "The purpose of many IMP tools is to reduce the work burden on the engineer, producer or musician, and to explore the extent to which various tasks can be automated" (p. 13). The subjects of IMP are Intelligent Assistants, Semantic Interfaces, Metering and Diagnostics, Interactive Audio. Moffat and Mark (2019) indicate three aspects of IMP when considering these subjects:

- Levels of Control: The engineer allows the system to do audio processing with different levels of controls. It can be achieved manually with data visualization, recommendation, independent modification of the parameters, or automatically.
- Knowledge Representation: Identification of solution for processing by given audio. This includes data representation and identification of engineers' habit

[1] Machine Learning techniques such as Neural Networks, Fuzzy Logic Systems, Evolutionary Algorithms.

on a task, or any procedural task defined by empirical formulas or data-driven approaches such as machine learning and deep learning systems.

- Audio manipulation: Acting with the knowledge of the presentation of the data, such as adaptive audio effects do.

Those aspects can be applied to every step of music production.

3. INSTRUMENT RECOGNITION

Instrument recognition studies started to appear in the early 1990s. Previously, the implementation of cepstral coefficients being successful in speaker identification systems motivated the usage of the same approach for instrument identification tasks "since a musical instrument can be viewed as a source plus resonator similar to a human speaker" (Brown, 1998, p. 1890). Instrument identification studies examine the audio file sources in two main categories as monophonic and polyphonic. A monophonic source audio file, also known as single-sourced or isolated-sound, contains only a single instrument during the whole recording. On the other hand, a polyphonic audio source comprises more than one instrument that sound simultaneously. Generally, the instrument identification studies were concentrated on single-noted solo recordings taken from audio libraries such as MUMS[2] and UIOWA[3]. However, for traditional (non-western) instrument recognition studies and early polyphonic-sourced recognition experiments, researchers consulted some custom libraries in the course of time. It is observed that the identification studies for polyphonic-sourced instruments gained importance after 2000.

3.1 Instrument Recognition Literature

One of the earliest studies in the monophonic instrument recognition model was presented by Kaminskyj and Materka. In this model, the classification performance of Artificial Neural Network (ANN) and Nearest Neighbor Classifier (NNC) algorithms were tested for a dataset that consists of RMS energy envelopes of four non-synthesized instruments (guitar, piano, marimba, accordion). The note range of the samples was limited (between C4-C5) in five different volumes. NNC showed slightly better performance (100% and 98.1% in different setups) over the ANN (97.7%) under the experimental conditions (1995). An extended version of this experiment was also

[2] McGill University Master Samples.
[3] University of Iowa Musical Instrument Samples.

presented by Kaminskyj and Voumard (1996). This time, nineteen different orchestral instruments from the MUMS library were examined with broad low-level audio features: amplitude envelope, constant Q frequency spectrum, spectral onset asynchrony, spectral onset peak position asynchrony, harmonicity/overtones, brightness, and articulation. According to the writers, these seven features and the classification methods provide successful results.

Martin and Kim (1998) present a pattern recognition technology with a taxonomic (hierarchal) approach for instrument recognition. In the taxonomic approach, instruments are divided into families such as strings, plucked, winds, etc.; then, they are classified individually. The auditory model is called a "log-lag correlogram" that measures perceptually salient acoustic features. These features were taken from 1023 individual tones within the overall pitch ranges for 15 orchestral instruments. The dataset was created with string, woodwind and brass families examined with Gaussian-based classification models. The attained identification accuracies were 70% and 90% for individuals and family classifications, respectively. There are two significant results. First, humans recognize objects taxonomically because of that a computer system will benefit from this approach. Second, timbre related acoustic properties in literature are helpful for instrument recognition.

Brown (1998) adapted the speech identification technique into instrument identification. The related speech identification system had been proposed before by Reynolds and Rose (1995), the idea behind that an instrument can be a source plus resonator as a human voice. The system was comprised of a Gaussian Mixture Model with Cepstral Coefficients. In this experiment, two wind instruments (oboes and saxophones) were used for the identification task. The result shows that the method can be used for instrument recognition.

Eronen and Klapuri (2000) presented the pitch independent musical instrument recognition method. This work aimed to build a more robust instrument recognition system with the guidance of the previously suggested recognition works. Cepstral and temporal features were combined to achieve this aim. 1498 solo tones and 30 orchestral instruments with different articulations were included in this research. Cross-validation with 70%/30% splits of train and test data were used. As a result, 94% accuracy for instrument family and 80% for individual instruments were achieved.

According to the paper, the hierarchal structured system is beneficial if more instruments with broad datasets are needed to be processed.

Wieczorkowska et al. (2003) highlighted the intriguing of automatic information extraction and explored how the sound presentation can be optimized by using MPEG-7 descriptors and their combinations in musical instrument recognition tasks. The multimedia data that provides title, time author, and similar meta-data information were not enough to estimate content-based searching such as the tune, segments, and instruments being played. Besides, there are no feature extraction algorithms available in the MPEG-7's multimedia content description interface. Therefore, the rough sets-based analysis method was applied to some patterns of observed features, and they were analyzed with the KDD (Knowledge Discovery in Databases) training data analysis method. That was a novel approach that applied data mining methods for particular audio samples.

Lee and Chun (2002) presented a feasible HMM (Hidden Markov Model)-based instrument recognition method. The algorithm divides the audio into its sinusoid and noise components; then, extracts amplitude from the sinusoid. Three instruments (flute, oboe, and violin) from MUMS and 40 samples for each were taken. With the three types of spectral audio features (amplitude, interpolated amplitude, and cepstrum), this method provided 70 % classification accuracy.

Herrera-Boyer et al. (2003) presented a detailed review about the automatic classification of (isolated) musical instruments with relevant audio features. They divided instrument identification into two main approaches as perceptual and taxonomic (like a bright snare, nasal violin, etc.). The related research reviews and features were provided in that paper for both perceptual and taxonomic classification ideas. Further investigation is required; yet the combination of perceptual and taxonomic approaches was stated in the conclusion.

Eronen (2003) proposed an instrument classification system of isolated notes or drum samples. The system uses Mel-frequency cepstral coefficients (MFCC) and first derivatives (ΔMFCC) features, then HMM (Hidden Markov Model) was used for feature distribution. The dataset consists of (isolated notes) 27 western orchestral instruments and drums. The feature vector was chained with ICA (independent

component analysis) and then trained with HMMs. It was observed that these processes improved the accuracy.

Costantini et al. (2003) highlighted the importance of single musical sources for source separation and music transcription systems. While experimenting, the preferred instruments were violin, clarinet, flute, oboe, saxophone, and piano. The FFT, QFT (Q-constant Frequency Transform), and cepstrum coefficients were used as audio features. The classification method was Min-Max Neurofuzzy Networks (generalized and classical) and performed adaptive resolution training algorithms (ARC, PARC, and GRAPC) to provide a completely automatic training for the classifier. According to the researchers, a combination of the FFT-based features, GPARC, and GMM model (mainly for uncritical training time) performed best.

Fanelli et al. (2004) used cochleagram as a feature selection technique with the deployment of LSTM (long short-term memory), which was introduced as a new recurrent neural network classifier. The dataset had 500 audio samples (70% for training and 30% for testing) and three instrument families (12 orchestral instruments from MIT media Laboratory Machine Listening Group). The proposed system showed 80% accuracy with the available dataset.

Kaminskyj and Czaszejko (2005) developed a mono-source instrument recognition system by combining investigated audio features up until then. The obtained audio features were cepstral coefficients, constant Q transform frequency spectrum, multidimensional scaling analysis trajectories, RMS amplitude envelope, spectral centroid, and vibrato. This paper explains that an instrument classification system can achieve 90% or more accurate results when providing sufficient features yet accentuates that attaining a better model generalization is difficult.

Kitahara et al. (2005) remarked on the objectives of accurately recognizing instruments among polyphonic recordings as feature variations, timbral pitch dependency, and musical context. In this paper, the first issue focused on a solution described as a *feature vector template*. The second issue was pitch dependent model, which represents the pitch dependency of the features as a timbral function of fundamental frequency (F0). The third issue was to solve the temporal continuity of the melodies by calculating the probabilities of each note regarding the probability of neighbor notes. The proposed method attained 85.8 % accuracy for trio music and 91.1%

accuracy for duo music. Afterwards, Kitahara et al. (2006) proposed a new polyphonic instrument recognition model called “Instrogram” without onset detection and F0 estimation. This time, the accuracy for trio music was 97.1% as maximum.

Essid et al. (2006) introduced a pairwise classification approach for solo instrument recognition. The dataset comprised audio excerpts for ten instruments from CDs and educational materials. In this study, besides the well-known audio features (temporal, cepstral, spectral, amplitude modulation), two new ones (octave band signal intensities and ratios) were examined by a Genetic algorithm (GA). An inertia ratio used feature space projection (IRMFSP) was preferred as feature selection algorithms to attain the most efficient discriminative feature subsets for among the instrument classes. Then, two types of classification methods (GMM and SVM) were conducted among these feature subsets by one versus one comparison. With larger observations and RBF kernel, SVM provided the best accuracy rate (93%). This study was one of the very extended studies at that time.

Wicaksana et al. (2006) also introduced a pairwise strategy for instrument recognition for the piano, violin, cello, piccolo, flute, and xylophone groups, with the classifier as a neural network. Primary audio features comprised temporal, spectral, and cepstral features and secondary features were attained from them. The composite network system shows 93.75% accuracy over the single network system (87.5%).

Leveau et al. (2007) presented a novel approach for polyphonic instrument recognition in ensemble music (solo, duo, trio and quartet). The audio sources comprise artificially mixed six ensemble instruments (bassoon, cello, clarinet, flute, oboe, viola and violin). This experiment was performed with a ten-second mid-term segmentation algorithm that approximates decomposed signal harmonic atoms of audio recordings so that the saliences of the instrument can be derived from the signal representations. According to the results, the method fails when the number of individual instruments increases in the ensembles

Even though the studies for instrument recognition mainly focus on western orchestral audio recordings, they also have been implemented for some traditional instruments. Yu et al. (2008) studied the song-level and segment-level identification approaches while estimating dominant instruments in polyphonic recordings of Chinese musical instruments (14 from 4 different families). GMM provided accuracies were 85% and

89% for those approaches, respectively. Also, it states that MFCC was not sufficient for musical instrument recognition.

Gunasekaran and Revathy (2008) proposed a fractal dimensions segmentation method to classify instruments. 85 audio features (temporal, spectral, and perceptual) were extracted from the ten different Indian instruments. Then, they were classified with k-nearest neighbor (KNN) and multilayer perceptron (MLP) algorithms; and it was observed that the fractional segmentation improved the accuracy by about 6% over local energy-based segmentation.

Burred et al. (2009) stated that the envelope of multiple sounds in an audio file is problematic during the audio feature extraction because it requires multi-pitch estimation for polyphonic instrument recognition tasks. The proposed system focuses on the amplitude evolution of the partials and compares them with pre-defined time-frequency templates to solve this problem. The best performances were 79.77% for two-voice, 77.79% for three-voice and 61.40% for four-voice polyphony.

During the years, studies mainly focused on the capabilities of audio features and classification algorithms, and some side works seemed to be needed to improve instrument recognition methods. These works range from refining audio datasets to eliminating drawbacks of some audio features. For example, Livshin and Rodet (2009) proposed an outlier detection algorithm for eliminating poorly recorded or incorrectly labelled samples in a database like MISDIN[4]. Later on, Gururani et al. (2019) highlighted that musical instrument recognition systems strongly depend on datasets and investigated an attention mechanism for a poorly labelled sound in a dataset like OpenMIC. The mechanism improved the overall performance of the classifier without extending the model. Morvidone et al. (2010) state that MFCC feature is useful but affects time scales during extraction for musical signals. OverCs (combine a compact description of MFCC) and SparCs (sparse representation of MFCC) features were proposed to overcome the immanent constraint of MFCC.

Wieczorkowska et al. (2011) studied the identification of the dominant instrument in audio mixes. The audio mixes were formed by adding basic signals and noises to the isolated sounds of 14 instruments from the MUMS database. An extensive feature

[4] Musical Instrument Sample Database of Isolated Notes

vector was available (219 features). It was observed that adding the extra noises and signals into the sounds in the training part of the dataset increased the classification performance compared to the isolated sound-only training dataset experiment result.

Bhalke et al. (2011) proposed a DTW (Dynamic Time Warping) method for musical instrument recognition tasks. DTW is an algorithm that finds similarities between time series by calculating a minimum distance. MFCC and its variants were audio features for this experiment. Instruments (Piano, Guitar, Violin, Synthesizer, Harmonica, and Accordion) regarded with 840 isolated notes were taken and divided as half for training and half for testing. It was observed that DTW improves recognition accuracy.

Azarloo and Farokhi (2012) compared MLP algorithm performance with the k-NN classifier using a dataset that has 296 audio features for seven different instruments. As a result, MLP performed better at 93.36% accuracy.

Eric et al. (2012) proposed a hierarchical classification method with a dynamic feature vector for instrument recognition. This was a method for feature vector optimization with a sequential backward selection (SBS) algorithm. 6698 different audios from RWC Music Database that consists of isolated notes with different dynamics for 29 single instruments were used to create a dataset with 38 audio features. Cross-validation was 90% for training and 10% for testing. The hierarchical classification rate for individual instruments was 88.32% and 94.74% for instrument families with the SBS algorithm by using k-NN. The hierarchical classification score performed better (2%) compared to the reference system.

Zlatintsi and Maragos (2013) introduced a non-linear method based on Mandelbrot's fractal theory to analyze signal irregularities in multiple time scales. The fractal theory was used to estimate the power spectrum in music and speech applications. This method uses MDF (multiscale fractal dimension) profile as a short time descriptor for seven instruments (1331 single notes) and its capability was compared with MFCC by GMMs and HMMs classifiers. As a result, this method provided up to 32% error reduction in the instrument classification experiments.

Giannoulis and Klapuri (2013) provided an extensive literature review for polyphonic sourced instrument recognition and proposed a missing-feature algorithm method that does not require sound separation or multi-pitch algorithms. The audio files, consisting of 10 different instruments, were taken from two different sound libraries (RWC,

MUMS). The log-energy differences feature was also introduced. Compared with the reference method that used MFCC and classification made with Bayesian (GMMs), the proposed method with different masks (oracle, heuristic, and max probs based on spectrogram factorization) performed well.

Diment et al. (2013) compared supervised and semi-supervised learning (SSL) techniques for instrument recognition. The audio recordings were taken from the RWC database as 4912 recordings in 9 different instruments. Only the MFCC features were used, and the EM-based SSL technique performed better by 16%.

The study of Zhang et al. (2013) presented a physiological model for the human auditory system. This system is also known as the bionic auditory system for instrument recognition mimicking the inner ear's cochlea structure and auditory cortex. Seven musical instruments (243 sound) were classified by self-organizing mapping neural network (SOMNN), and a success rate of at least 75% was achieved.

Kursa and Wieczorkowska (2014) introduced multi-label ferns and their application for instrument classification. The dataset was created with jazz recordings MUMS, IOWA, and RWC by extracting mainly MPEG-7 low-level features for this study. The multi-label ferns provided faster, and more accurate performance compared to the binary random ferns. This algorithm was recommended for mobile devices for instrument recognition and similar applications.

Bhalke et al. (2014) introduced a classification method for string instruments consisting of monophonic, isolated sound sources. The method uses fractional Fourier transform (FRFT) based on MFCC features. The system performance was compared with the conventional features (MFCC, Timbrel, Wavelet, and Spectral features). The classification method was Linear Discriminant Analysis (LDA) which performed with 94% accuracy.

Kothe et al. (2016) compared classification performances of k-NN and SVM algorithms with different settings for individual instrument recognition. Temporal, spectral, cepstral and wavelet features (31 features) were extracted for ten musical instruments. As a result, using the weight factor method, the SVM (with the exponential kernel) shows 90.3% accuracy over KNN, which has 73% accuracy.

Lostanlen and Cella (2016) used Deep Convolutional Networks (DCNs). This was for instrument recognition in three strategies: temporal, time-frequency kernels and a

linear combination of the time-frequency kernels in one octave (similar to Shepard pitch spiral). Training data (from MedleyDB) and evaluation data (custom music collection) were different for this experiment. Combining them into a single deep learning architecture shows the best accuracy result.

Toghiani-Rizi and Windmark (2017) compared the time and frequency features of different characteristics of 8 different musical instruments with the ANN system. These characteristics were attained by extracting the features from some parts such as the whole sound, attack part, the whole sound with the attack part excluded, initialization frequency spectrum from 100 Hz, and the following 900 Hz of the spectrum. The feature vector that included whole samples constrained with 1-1000 Hz range of the spectrum provided 93.5% accuracy.

Maliki and Sofiyanudin (2018) proposed an ANN system called Learning Vector Quantization (LVQ) for musical instrument recognition. The instruments were grouped as Aerophone, Electrophone, Chordophone, Idiophone, and Membranophone. The feature vector has only MFCC features, and the LVQ classification showed 94.8% accuracy with the dataset.

Rajesh and Nalini (2020) asserted a new approach for emotion recognition using musical instrument classes by recurrent deep learning (RNN) method. Four instruments (string, percussion, woodwind, and brass) are grouped to prompt the emotions (happy, sad, neutral, and scared). The features for the dataset were MFCC, CENS (chroma energy normalized statistics), chroma STFT (short-time Fourier transform), some spectral features, and ZCR. MFCC feature with RNN provided the best classification rate of 89.3%, compared to the other machine learning and feature combinations.

Roy et al. (2020) represented another application of ANN in musical instrument recognition. This method is based on frequency domain analysis to extract distinct features in the transition part of the audio signals. Its name is TMIR (transient length instrument recognition), and a restricted range of frequency domain is analyzed rather than the whole spectrum. The algorithm analyzes the spectral content for the maximum energy part of the onset. The dataset was examined with an ANN-based model for five different instruments, and 98.22% accuracy was attained regarding the true fixed transient length of the audio files.

Mahanta et al. (2021) examined twenty orchestral instruments from the London Philharmonic Library. The audio samples were single notes with different dynamics and playing techniques. The feature vector consisted of MFCC and provided 97% accuracy for instrument recognition by ANN.

Garcia et al. (2021) explained an instrument recognition approach deploying hierarchical relationships by conducting a few-shot learning setup for improving a successful classification model for unseen instruments. The audio files were taken from the MedleyDB library and grouped as 24 families; then, Mel spectrogram features of solo music and vocal recordings were extracted. This method showed better performance over the non-hierarchical method.

Ezers and Doroshin (2021) introduced a CNN based real-time musical instrument recognition android application. The system uses the Mel-spectrogram of a wide range of instruments from the IRMAS dataset. For improvements on the accuracy result of the software, some strategies were presented.

Li (2021) highlighted the advantages of musical instrument recognition solutions in music education by presenting a method for piano education with the assistance of deep learning technologies.

3.2 Sound, Signal and Fundamental Concepts of Digital Audio

The occurrence of the sound depends on the type of source, medium (transmission or storage), and its interpretation by a receiver. After the stimulation of a sound source, vibrations take place and are transmitted as sound waves in a physical environment. If the sound is transmitted electrically, then it is called a signal. The signal can be analog (continuous) or digital (discrete).

Digital systems have many advantages over analog systems regarding easy transmission and storage. On the other hand, digital systems provide further processing and manipulation capabilities on sound; in fact, digital signal processing (DSP) is a core discipline of that field.

3.2.1 Sound

In terms of physics, sound occurs when a source is vibrating, and this vibration is transmitted through a medium by compression and decompression of air molecules to

a receiver. The simple harmonic motion describes the formation of a simple sound wave. Sound waves have a period that describes the frequency which is the repetition of the motion in a second. The motion of a pendulum is an example of simple harmonic motion. After a mass (m) is pulled back to the distance (a) from the equilibrium point (a=0) and released, it sways until its energy has damped, and this is called a simple harmonic motion (Figure 3.1, A). The amplitude of the motion also equals the maximum distance from the balance point. This movement can be explained as a point that completes one cycle through a unit circle over time (Figure 3.1, B). The cycle duration is represented by the period (*T*), and the number of cycles in a second are represented by the frequency (*f*). Hence, the relationship between frequency and the period is in equation 3.1.

$$T = \frac{1}{f} \tag{3.1}$$

For a sound wave, another essential property is its phase. Phase indicates the starting point of an oscillation for a reference point in time. In Figure 3.1 (B), the sound oscillation starts at phase 0 and completes its period at the time *t*.

The simple harmonic motion clearly describes a basic sound, but sounds have more complex structures in nature. Combining more than one basic sound with different phases, amplitude, and frequencies creates a complex sound.

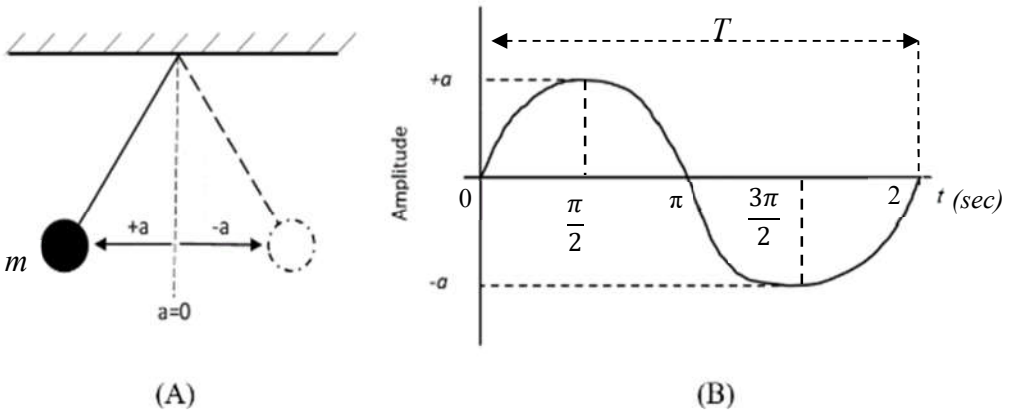

Figure 3.1: Basic harmonic motion: a pendulum (A) and its function (B) in duration t.

3.2.2 Signal

Besides the physical environment, sound waves are also represented as signals in analog and digital domains. “A signal is defined as any physical quantity that varies with time, space, or any other independent variable or variables. Mathematically, we describe a signal as a function of one or more independent variables” (Proakis & Manolakis, 1996, p. 2). However, an audio signal cannot be formulated as a regular mathematical function because it is a periodic signal which continuously repeats itself at a particular time duration.

Amplitude (A), frequency (f), and phase (θ) are variables for an audio function $y(t)$. A sinusoid is a fundamental type of audio signal, and its is given in equation 3.2.

$$y(t) = A\,sin(2\pi ft + \theta t) \tag{3.2}$$

In this formula, f is the fundamental frequency (f_0). The complex sound is formulated by the summation of simple sounds. In equation 3.3, N stands for the number of simple sinus waves and i indicates the current one.

$$\sum_{i=1}^{N} A_i(t) sin\,(2\pi f_i(t)t + \theta_i(t)) \tag{3.3}$$

This formulation explains the harmonic content of a sinus-based complex sound. If the f_0 is fundamental frequency, its multiplication with integers (1, 2, 3…, n) gives the harmonics (f_n) of the complex sound.

There are two types of signal system definition in terms of time. “Continuous-time signals are defined along a continuum of time and are therefore represented by a continuous independent variable. Continuous-time signals are often referred to as analogue signals. Discrete-time signals are defined at discrete times, and thus, the independent variable has discrete values; that is, discrete-time signals are represented as sequences of numbers.” (Oppenheim & Schafer, 2009, p. 9)

3.2.3 Basics of digital audio

Digital signal processing is essential for modern communication technologies. There are many advantages of digital systems over analog systems. The advantages of digital systems are, the compatibility of different conditions, accurate control, easy storage

and transmission, implementation of sophisticated signal processing techniques, and low cost (Proakis & Manolakis, 1996, p. 6).

The sampling theory and quantization are two fundamental concepts from analog to digital conversion and vice versa. Sampling is a discretization process in which a certain number of amplitude points are taken from an analog signal in time. Figure 3.2 shows the density of sampling points of 440 Hz signals sampled with 44100 Hz, 22050 Hz, 11025 Hz, and 52512 Hz, respectively; hence, it can be observed that the quality of the audio signal reduces when sampling frequency (rate) decreases. The sampling theorem states that sampling frequency must be higher than twice the signal's highest frequency to prevent information loss when reconstructing the signal. "If the signal contains frequency components higher than half the sample rate, these components will be mirrored back, and aliasing occurs." (Lerch, 2012b, p. 10).

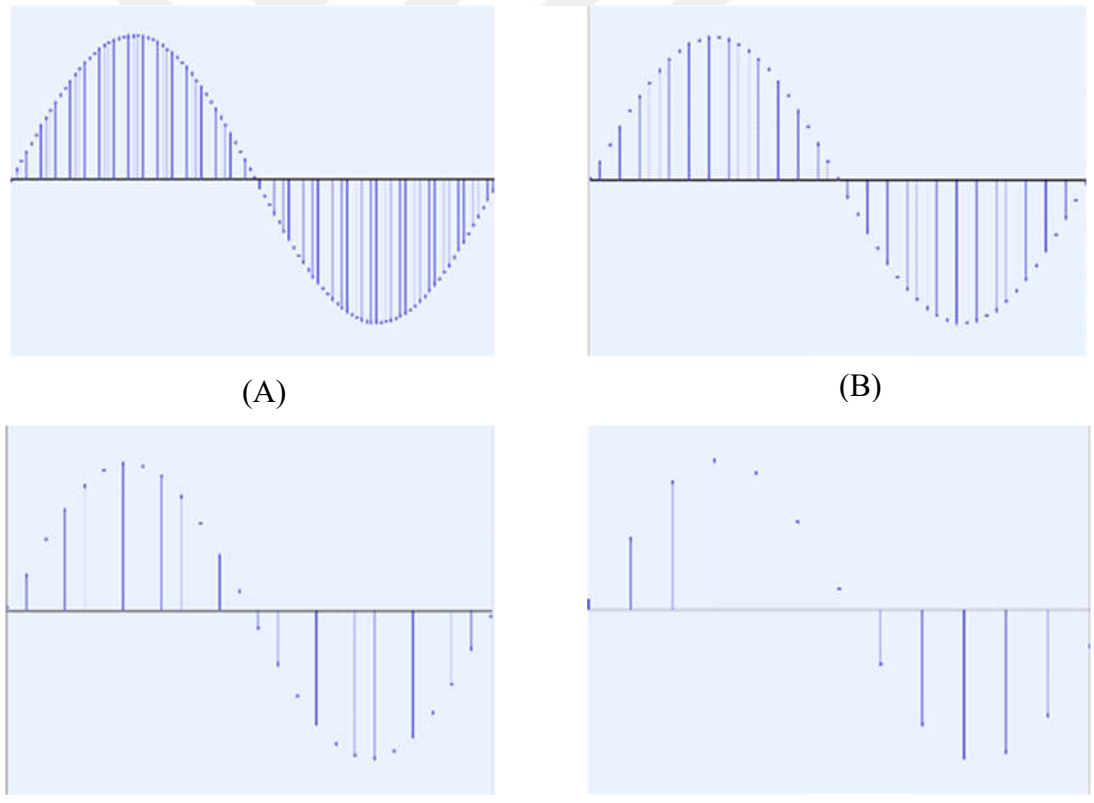

Figure 3.2: One period of 440 Hz sinus signals with different sampling frequencies. The sampling rates of the figures are 44100 Hz (A), 22050 Hz (B), 11025 Hz (C), and 5512 Hz (D). Decreasing the sampling frequency reduces the number of sample points which affect the quality of the signal.

Each sampling point has an amplitude value, and during the sampling process are rounded to the nearest amplitude values, which is determined by the bit depth of the signal. Bit depth is also called word length and is typically preferred as 16 or 24 bits, and they are represented by integer or floating points data types. The 16-bit word length provides -32768, +32767 predefined amplitude values on positive and negative axes (Figure 3.3). However, during amplitude values of the input signal do not coincide with that amplitude points, they are rounded to the nearest predefined steps. The difference between the actual amplitude value of the original signal and its quantized value gives the quantization error.

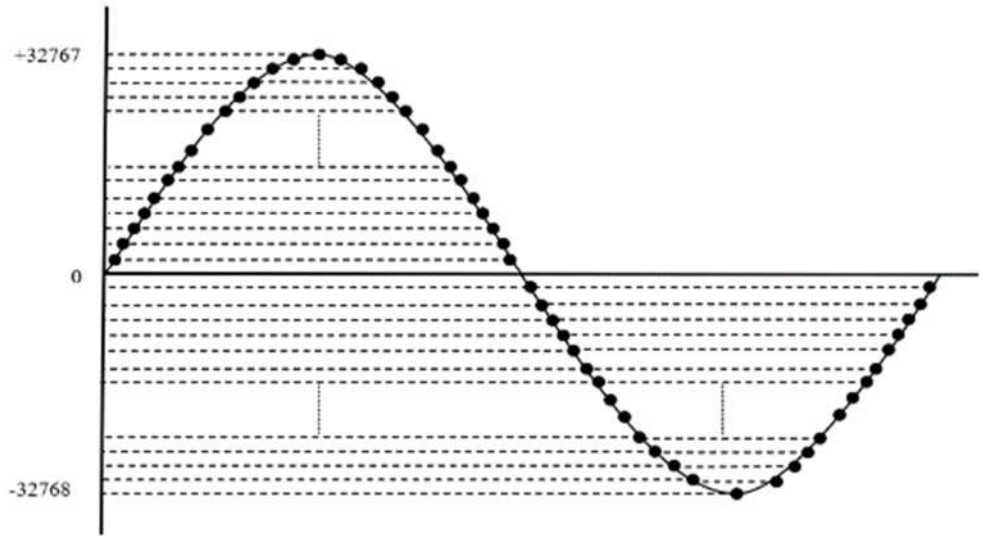

Figure 3.3: In this figure each point represents the available quantization steps in 16-bit depth. When analog to digital conversion, amplitude values between those predefined points rounded to the nearest quantization points.

3.2.4 Digital signal processing

The digital signal processing systems aim to solve transmission, storage, and manipulation (like compression) problems of discrete signals.

For continuous (analog) signals, the Fourier transform is a mathematical formula that converts a time-domain signal into the frequency domain by representing the signal's component as real and imaginary parts. "The basic mathematical representation of periodic signals is the Fourier series, which is a linear weighted sum of harmonically related sinusoids or complex exponentials" (Proakis & Manolakis, 1996). The Fourier transform allows acquiring the spectral components of a complex signal and processing them individually.

The DFT (Discrete Fourier Transform) is essential for digital signal processing tasks. “The DFT is itself a sequence rather than a function of a continuous variable, and it corresponds to samples, equally spaced in frequency, of the DTFT of the signal.” (Oppenheim & Schafer, 2009, p. 623).

In practice, DFT is applied by dividing the signal into short-time segments. This method is called STFT (Short-term Fourier Transform) and it “can be seen as multiplying an infinite time signal with a window function that equals zero outside the time boundaries of interest.” (Lerch, 2012a, p. 192). The typical segment length of STFT is between 10-20 milliseconds, each convoluted by a window function. A window function decreases the amplitude value of each segment edge (side lobes) to zero; yet, the amplitude value stays at maximum at the center (main lobe). Rectangular, Bartlett, Hamming, Blackman-Harris are some examples of the types of window functions.

3.3 Audio Content Analysis

The audio content analysis (ACA) methods are used to obtain musical or non-musical information from audio signals. “The information to be extracted is usually referred to as metadata: it is data about (audio) data and can essentially cover any information allowing a meaningful description or explanation of the raw audio data. The metadata represents (among other things) the musical content of the recording” (Lerch, 2012a). The information can be perceptual, musical, or technical and interpreted according to the purpose of the research or application domain. Therefore, the ACA is a multi-disciplinary research field that integrates digital signal processing methods with musicology, music theory, music psychology, psychoacoustics, audio engineering, and similar knowledge with pattern recognition and machine learning methods. This field urges the creation of tools for organization, visualization, intelligent audio processing, and identification of audio data.

A general workflow of an ACA is shown in Figure 3.4. The “interpretation” step describes any multivariate statistical technique such as (shallow) machine learning, deep learning, or even artificial intelligence in this scheme. Depending on the task, the result can be a class name or a number.

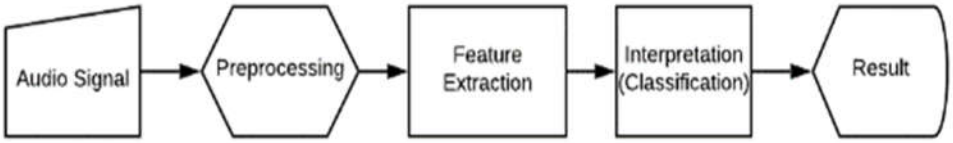

Figure 3.4: A general workflow of an ACA system.

3.3.1 Audio pre-processing

Before starting the extraction process, audio pre-processing is a common practice. It consists of several operations such as sampling rate conversion, DC removal, stereo to mono conversion, silence removal, normalization, filtering, and even segmentation. "The motivation for this pre-processing step is to reduce the amount of audio data to be analyzed by omitting unnecessary information or minimize the impact of unwanted information on the extracted features" (Lerch, 2012, p. 33).

3.3.2 Audio feature extraction

The feature extraction is a process that reduces the data size of the audio file substantially; therefore, a more relevant and meaningful representation of the audio data emerges. This process occurs at a predetermined length of the frame. As a result of the extraction, low-level features (also called instantaneous or short-term features) are calculated at each window frame. The outcome of the low-level processing is not directly interpreted by humans, but further statistical interpretation of those features leads to determination of high-level features, which means a more meaningful representation for humans.

3.3.3 Low-level (short term) processing

The low-level feature extraction is a block-based process. According to this process, the audio file is divided into pieces (blocks), and feature extraction is carried on each of them. Typically, the duration of block size is between 128-1024 samples or equivalent time duration, which is determined by the division of block size with the sample rate for a given audio file. Each audio block is processed with a window function.

The feature calculation takes place in two domains, time domain and frequency domain. The time-domain features materialize after the summation of successive frame-based sequences, so they are determined directly from the audio file. On the

other hand, the FFT (Fast Fourier Transform) is necessary to acquire frequency domain features. The audio features can be calculated by linear or non-linear magnitude and spectrum scales (such as Bark, Mel). Mostly, non-linear scales perform better because the non-linearity of the human auditory system.

Giannakopoulos (Giannakopoulos, 2015) described the widely used audio features as zero crossing rate, energy, entropy of energy, spectral centroid, spectral spread, spectral entropy, spectral flux, spectral roll-off, MFCC, chroma vector, and harmonic ratio. Table 3.1 shows descriptions of those features.

Table 3.1: Widely used time and frequency domain audio features.

	Audio Feature	Descriptions
Time Domain	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
	Energy	The sum of squares of the signal values, normalized by the respective frame length.
	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as energy measure of abrupt changes.
Frequency Domain	Spectral Centroid	The center of gravity of the spectrum.
	Spectral Spread	The second central moment of the spectrum.
	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
	Spectral Roll-Off	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
	MFCC	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the Mel-scale.
	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

3.3.4 Mid-level processing

Mid-term windowing (also called texture windowing) is a common practice to "exhibit homogeneous behavior with respect to audio type and it, therefore, makes sense to proceed with the extraction of statistics on a segment basis" (Giannakopoulos & Pikrakis, 2014, p. 62). Kitahara (2010) explained the necessity of mid-term representation to attain melodic, harmonic, timbral, and rhythmic aspects of an audio source. In the mid-term processing "the audio signal is first divided into mid-term windows (segments), which can be either overlapping or non-overlapping. For each segment, the short-term processing stage is carried out and the feature sequence from each mid-term segment is used for computing feature statistics (e.g., the average value of the ZCR)" (Giannakopoulos, 2015). Generally, the duration of the mid-term segment is between 1-10 seconds, but it depends on the task.

3.4 Audio Dataset

The content of a dataset varies from application to application; yet audio datasets generally consist of numerical data at each sequence. Therefore, each sequence of data holds a row, and each column identifies a specific feature. In addition, a dataset must have an extra column to indicate the target labels or values for supervised machine learning tasks. Table 3.2 presents a typical M x N dataset.

Table 3.2: MxN dimensional dataset.

Index	Observations					Targets
0	Feature 1	Feature 2	Feature 3	…	Feature N	Class/Value
1	Value 1_1	Value 1_2	Value 1_3	…	Value 1_N	Class/Value
2	Value 2_1	Value 2_2	Value 2_3	…	Value 2_N	Class/Value
3	Value 3_1	Value 3_2	Value 3_3	…	Value 3_N	Class/Value
…	…	…	…	…	…	…
M	Value M_1	Value M_2	Value M_3	…	Value M_N	Class/Value

3.5 Machine Learning and Classification

Machine learning is a computer algorithm that predicts input data by a mathematical model with provided training data. According to Saleh (2020), "Machine learning

(ML) is a subset of Artificial Intelligence (AI) that consists of a wide variety of algorithms capable of learning from the data that is being fed to them, without being specifically programmed for a task." (p. 3). These algorithms are beneficial when input data has a complex structure that cannot provide a meaningful result by conventional mathematical formulas. The machine learning systems rely on statistics and pattern recognition methods while classifying, discriminating, or predicting input data. These systems are applied in many different research fields such as engineering, medical, biology and the like. The size of the data may reach enormous quantity according to the application domain. In that kind of situation, the data need to be simplified for further usage. This simplification is called Data Mining.

3.5.1 Probability in machine learning

There are three mathematical aspects to data modelling: Function Approximation, Optimization, Probability and Statistics. Function Approximation stands for the mathematical relationship between variables without knowing the function entirely. Optimization aims to find the best mathematical model for a given datum by calibrating or fine-tuning the model. Probability and Statistics are necessary when approaching inherently uncertain data to make accurate predictions (Kroese et al., 2019).

"Probability theory provides a consistent framework for the quantification and manipulation of uncertainty and forms one of the central foundations for pattern recognition." (Bishop, 2006, p. 12). The theory of probability is interpreted in two main categories as frequentist and uncertainty. Frequentist can be exemplified by observing whether a flipping coin is head or tail at a long-term rate. In contrast, the Bayesian interpretation is proper when the probabilistic model is based on uncertainty, and the event does not occur for long term frequencies. The melting of the polar ice cap by 2050 CE is an example of Bayesian probability (Murphy, 2012).

Bishop (2006) explains the probability theory by providing a simple frequentist example. In this example, two different types of fruits (apples and oranges) are in two boxes (red and blue). The red box has two apples and six oranges, but the blue box has one orange and three apples. The percentage of picking the red box is 40% while it is 60% for the blue box (those boxes may be considered as classes in machine learning). The fruits are randomly picked from one of those boxes. The boxes are denoted by X

and the possible values are r or b (red or blue). Y denotes the fruit names and takes two values a or o (apple or orange). We write these probabilities as p(X = r) = 4/10 and p(Y = b) = 6/10 so that they normalize between 0 and 1. As a consequence, the probability of the X (baskets) and Y (fruits) are given in Table 3.3.

Table 3.3: Example for the probability of the X (baskets) and Y (fruits).

	Number of apples	Number of oranges	probability
Red Basket (r)	2	6	p(a | r) = 1/4 p(o | r) =3/4
Blue Basket (b)	3	1	p(a | b) = 3/4 p(o | b) = 1/4
Overall probability	11/20 p(a | r) p(r)+p(a |	9/20 p(o | r) p(r)+p(o | b)p(b)	Equals 1 in total

The system which consists of two parameters (boxes and fruits) creates the joint probability, p(X, Y), and conditional probability, p(Y|X). The rules of probability theory are given as sum rule and product rule (below) for X (box) and Y (fruit) components. p(X) stands for probability of X (so the box) and is also called marginal probability because the other variable (Y) is excited (Table 3.4).

Table 3.4: Sum rule, product rule, and Bayesian probability theory.

Sum rule (marginal probability)	$p(X) = \sum_{Y} p(X,Y)$
Product rule (joint probability)	$p(X,Y) = p(Y\|X)p(X)$
Bayesian probability	$p(Y\|X) = \frac{p(X\|Y)p(Y)}{p(X)}$ $p(X) = \sum_{Y} p(X\|Y)p(Y)$

The Bayes theorem is an essential part of pattern recognition and machine learning. The theorem has two types of probability: prior and posterior. If the probability result of X, p(X), is calculated before the knowledge of the probability of Y, this type of probability is named prior probability. However, if the probability result of X, p(X|Y), is calculated according to the specific value of Y, then this type of probability is called posterior probability.

The probability is also used to represent uncertainty parameters, (w), which is revised by new evidence for an event, D. Therefore, the event (D) and uncertainty (w) creating posterior probability is given in equation 3.4.

$$p(w|D) = \frac{p(D \mid w)p(w)}{p(D)} \tag{3.4}$$

3.5.2 Unsupervised machine learning

Unsupervised learning methods use density estimation which aims to find out the regulations in the input data; thus, the input value is grouped according to their similarities. This type of learning is called clustering. Clustering is useful and it performs better when reducing the bit-depth of an image by finding regularities of what is shown by that image rather than pixel level compression. Also, clustering is used in distinguishing a document style, words of a document, or DNA sequences in bioinformatics (Alpaydin, 2014). Saleh (2020) sorted clustering into four models as connectivity-based, density-based, distribution-based, and centroid-based. Also, search engines, recommendation programs, image recognition, market segmentation (customer experience) are some of the applications of this type of machine learning.

3.5.3 Supervised machine learning

Supervised learning "consists of understanding the relationship between a given set of features and a target value, also known as a label or class." (Saleh, 2020). As a result, classification and regression are two types of supervised machine learning tasks.

3.5.4 Classification and regression

A machine learning model has a prediction function (g) that accepts a vector (x) in its input and produces a result (y) (Figure 3.5). If the result of the function is a real value, this type of learning problem is called regression. On the contrary, if the result of the function produces a value among a categorical set of values, then this type of learning problem is defined as classification.

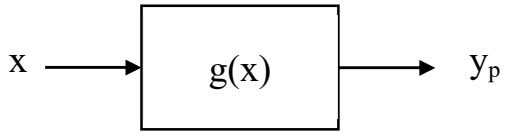

Figure 3.5: A machine learning model has a prediction function (g) that accepts a vector (x) in its input and produces a result (y).

3.5.5 Model validation procedure for supervised learning

The model evaluation starts by splitting the dataset into two (training and testing) or three groups (training, validating, and testing). Those sets are used during the development of a consistent machine learning model.

In the two-set scenario, the dataset evaluation occurs by splitting the dataset, and the lack of real-life data examples affects the hyperparameter tuning; so, the model tends to be biased. Therefore, the K-fold cross-validation technique prevents a biased result during the machine learning model evaluation. The dataset is split into a K number of pieces by this technique and data in each piece are randomly selected during the K time iteration. As a consequence, hyperparameters of the model are tuned in that way.

For the three-set scenario, the training set has been examined by many different machine learning models. The validation set allows hyperparameters tuning for each machine learning model. A loss function allows calculating the hyperparameters. "After running different configurations of hyperparameters based on the performance of the model on the validation set, a winning model is selected for each algorithm." (Saleh, 2020, p. 92). After training and validation, the model is examined by the testing set to find how the selected model is performed in real-life conditions.

The confusion matrix (Table 3.5) is a valuable tool to obtain accuracy, precision, recall, and F-score metrics in machine learning. The confusion matrix is a table that shows the ground truth (actual values) and prediction relationship of a model. While

rows (ground truth) represent the predicted values, columns show the actual values. Four types of indicators are deduced from a confusion matrix: True positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). TP gives correct and TN gives the number of false observations which are expected. However, FP (predicted as positive but false) and FN (predicted as negative but false) are two error indicators.

Table 3.5: A confusion matrix.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Model accuracy is measured by the summation of the TP and TN by dividing the number of instances (N) and indicates with which rate the model can classify the data correctly (equation 3.5).

$$Accuracy = \frac{TP + TN}{N} \tag{3.5}$$

The precision metric is measured by dividing TP by the summation of TP and FP. Although the precision originates from estimating the accuracy of binary problems, it is also applicable to multiclass problems (equation 3.6).

$$Precision = \frac{TP}{TP + FP} \tag{3.6}$$

Recall metric is similar to precision, but it is calculated by dividing TP by the summation of TP and FN (equation 3.7).

$$Recall = \frac{TP}{TP + FN} \tag{3.7}$$

Moreover, the F-score may provide a better understanding of a model. The F-score is the harmonic mean by the combination of precision and recall (equation 3.8).

$$F - score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (3.8)$$

3.5.6 Error analysis

Error analysis is necessary to create a generalized model. Calculation of Bayes error (human error), high bias, high variance and data mismatch are error checking concepts for formalizing solid learning models. High bias (also known as underfitting) describes the situation when the model cannot learn from the provided training set. High variance (also known as overfitting) occurs when the model is so deeply rooted in the details and outliners of the training set that it performs poorly with test set or unseen data. Finally, data mismatch occurs if training data distribution does not fit with the test data or validation data distributions (Saleh, 2020).

3.5.7 Support vector machine algorithm

The machine learning models are divided into two main categories as shallow and deep methods and the theory behind each of them is diverse. In this dissertation, after comparing the performance results with the same dataset, support vector machine algorithm was considered as instrument classification model. “The Support Vector Machine (SVM) algorithm is a classifier that finds the hyperplane that effectively separates the observations into their class labels.” (Saleh, 2020, p 143). For a binary classification problem, a hyperplane symbolizes the boundary between two classes. Indeed, the SVM algorithm can separate the new data according to the previously estimated hyperplane. The SVM is based on a binary (two-class) classifier. However, it can be generalized by one-versus-one or one-versus-the-rest methods to solve multiclass problems.

Figure 3.6 demonstrates a hyperplane and two support vectors that take place on the positive and negative sides of the hyperplane at equal distances.

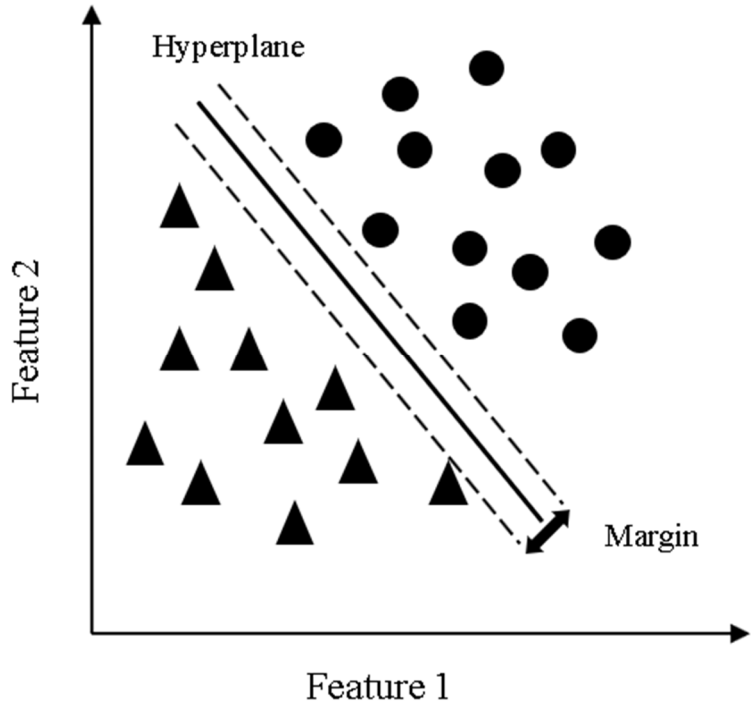

Figure 3.6: Demonstraits a (linear) SVM algorithm for binary classification. Triangles and circle are two different classes.

In order to estimate the right hyperplane, the SVM algorithm carries on two rules. Rule 1 designates finding the best hyperplane position where data are effectively separated. On the other hand, rule 2 (margin) specifies the maximum distance against the nearest data points from both sides (classes). A more significant margin distance (or minor margin error) increases the efficiency of the model. During the estimation of the hyperplane, the SVM algorithm aims to find the ultimate segregation between classes, and this is provided by finding minimum classification and margin errors. The classification error is calculated by the distance measurement of each data point from the boundary (hyperplane).

The SVM algorithm is a kernel-based classification method, which means the hyperplanes are customizable by different functions. The Scikit library allows using linear, polynomial, radial basis function (RBF), sigmoid, and custom kernels.

4. DEVELOPMENT OF THE MIX PREPARATION ASSISTANT

This section depicts the methodological structure of the study by the conceptualization and implementation of the mix preparation software. Before starting the software development, some preliminary decisions had to be made to determine the development platform and the DAW where the software would work. We determined the key features and presented a detailed GUI layout of the software by making some decisions over the software's capability, keeping in mind the fundamentals of the mix preparation approach. The design of the GUI and the functionalities of each part are explained in detail.

4.1 Daw Preference and Software Development Platform

Two decisions must be made before beginning the development of the mix-preparation assistant software. First, the software needs a host (DAW) to work coordinatively for practical reasons. However, commercial DAWs are proprietary software, and only a few allow controlling their functionalities by scripting languages. At that point, Reaper becomes a unique candidate for this dissertation research among the other workstations because its open-source structure provides comprehensive control over its functionalities in terms of software development. The control over the Reaper API is achieved by the ReaScript wrapper and supports EEL2, Lua, and Python scripts[5].

Python is a prominent and well-known programming language today. With its high-level, user-friendly syntax, portable and object-oriented language, it meets almost all industry requirements, such as gaming, cloud systems, and education. Python is also one of the essential tools for making machine learning applications, too. Hence, the proposed mix preparation solution in this study is revealed by the combination of Reaper and Python.

[5] For details, https://www.reaper.fm/sdk/reascript/reascript.php

4.1.1 The research workflow

In this study, previously, MATLAB was considered a research environment to investigate audio features and machine learning models for instrument recognition tasks. After gaining experience in this research field, the next step was to decide the implementation method of the audio feature algorithm and machine learning model into the mix preparation software. However, the implementation of the MATLAB code into a different programming environment is a cumbersome task. Also, C++ based MIR, or ACA libraries[6] (such as Essentia) are not practical for compilation needs and their dependencies for computer platforms. Because of those reasons, a Python-based machine learning library was preferred as a software platform for this dissertation research.

4.2 The Properties of the Mix Preparation Software

As mentioned in chapter 2.1, the definition of mix preparation has objective and subjective sides. Here, the proposed system has been restricted to only track organization which is rather the objective side of mix preparation. The obtained design concept of the software regarding the track organization demands of mix-preparation are given below.

1. The software should have an intuitional GUI plain and a simple layout.
2. The software should allow importing of audio files from the given multi-track project.
3. The software should have customizable audio groups (for further usage such as mute, solo, and other standard functionalities), audio folders, coloring, and names.
4. The software should create a Reaper project according to the user's preferences. Also, it should be able to clear the project so that users can start creating the project without instantiating the software again.
5. The software should do the given steps automatically, but users can interfere with the procedures before and after the project creation.

[6] A list of MIR Libraries are given in https://ismir.net/resources/software-tools/

6. The software allows users to create custom datasets and ML models for further usage.

4.3 The GUI of the Software

Natively Python has a GUI library called Tkinter and the software's interface was built only using this library. The GUI of the program is grouped into three main sections by Tkinter's frame widget marked as A, B, and C shown in Figure 4.1.

A) *Folder Importer*: After importing audio files from a multitrack project appears here.

B) *Instrument Groups*: This part lists ten individual subgroups and each has default names and colors. The imported audio files shown in *Folder Importer* section were assigned to the selected subgroup frames either manually or automatically thanks to the ML-based subgrouping function.

C) *Machine Learner Section*: This section allows to build custom datasets with a set of audio features besides some statistical properties which users can select each of them on purpose. This section also provides the opportunity to create an SVM-based supervised machine learning model with an audio dataset.

The first section is the Folder Importer (marked as A), which has two buttons and one list widget. The Select Folder button opens a dialogue box that expects users to select a multitrack project folder. The audio file importer only accepts audio files with .wav extensions but ignores other audio and non-audio formats from the selected folder. The included audio files from the Folder Importer list widget can be added one by one or as multiple (by holding ctrl or shifting from the keyboard) to any of the preferred instrument groups in the section named Instrument Groups (marked as B). Adding or discarding audio files between the Folder Importer and Instrument Group can be carried out by clicking the buttons (<, > symbols). The keyboard shortcuts make multiple file selections possible, even removing the audio files from an instrument group. The button Clear List resets the import list and all subgroup assignments.

Figure 4.1: The GUI of the mix preparation assistant software (MixPrep).

Ten subgroups exist in the Instrument Groups section (marked as B), and their names are Group A, Group B, Group C, ..., Group J as default. Besides the manual modification, subgroup names change automatically according to target names determined in the dataset if the user prefers setting the instrument family identification algorithm. In the bottom-right side of each instrument group window, there is an entry widget intentionally left as white on the group color modifier, and the current group name can manually change by clicking on this entry widget. For the sake of simplicity, the entry widgets' character limitation is 12 letters in length; yet white spaces are not allowed. Color modification is also possible for each subgroup. By clicking the colored part of the bottom of the current instrument group, a color picker dialogue box appears (Figure 4.2).

As stated in the properties of the software, it is expected to provide an intelligent solution for automatic subgrouping tasks. For this purpose, the GUI includes a dedicated *Machine Learner Section,* where individuals design their datasets and related instrument family recognition models. In that section, there are two buttons and one label widget. *Choose Model* button opens a file dialogue window that expects the user to select a machine learning model. After the user prefers a machine learning model, its name and path appear in the label widget. The model must be created by

MixPrep's machine learning module (shown in Figure 4.3) based on pyAudioAnaysis, the Python library.

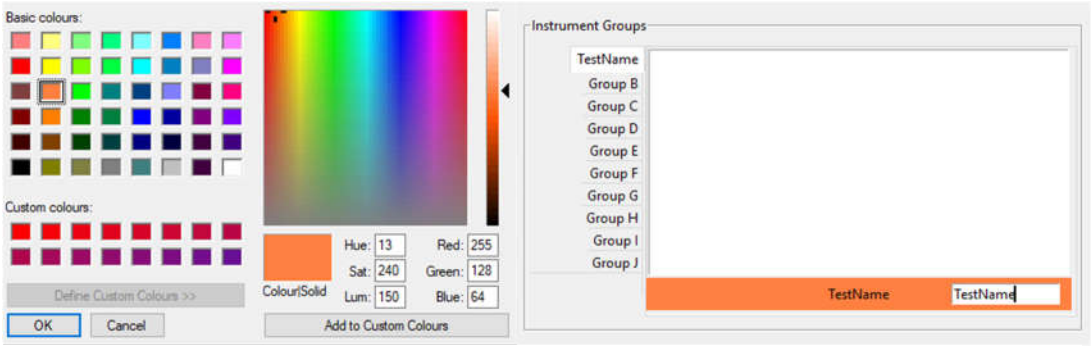

Figure 4.2: Color picker window and name modifications for the first subgroup.

After the user has imported a multitrack folder and organized the audio files into different subgroups manually or automatically, the subgroup modifications (names, colors, and contents) are ready to be transferred to Reaper. The software imports all selected audio files into the currently opened Reaper session and creates subgroups with provided or default names and channel group colors by clicking the Create Reaper Session button. It should be noted that MixPrep will not transfer the empty subgroups into the Reaper session. The button *Clear Reaper Session* cleans out the opened Reaper session so that users can start subgrouping again without reinitializing the MixPrep.

4.4 The Machine Learning Module

The machine learning module appears as a new window by clicking the *Machine Learner* button (Figure 4.3). Users create custom machine learner models and datasets with preferred audio feature sets and different statistics with this module. Due to the complexity of building an audio features extraction and ML pipeline, an open-source library called pyAudioAnalysis is preferred over the others because it allows the implementation of a wide range of applications from speech recognition to musical classification with many functionalities: feature extraction, classification, regression, segmentation, and visualization as one library (Giannakopoulos, 2015). The library already has the mean and standard deviation for mid-term statistical calculations for each column natively. However, previous experiences with MATLAB have proven that the presentation of median statistics for feature columns would be pretty helpful while the ML algorithm is working on the classification of the target columns.

Therefore, the library's function where mid-term feature statistics calculation happened is revised due to adding median statistics for the features columns in datasets. Again, since the library seemed deficient in dataset conservation and further manipulation, it is considered necessary to add the functionality of saving datasets as an excel file. Lastly, an audio excerpt algorithm was added to efficiently deal with longer audio files.

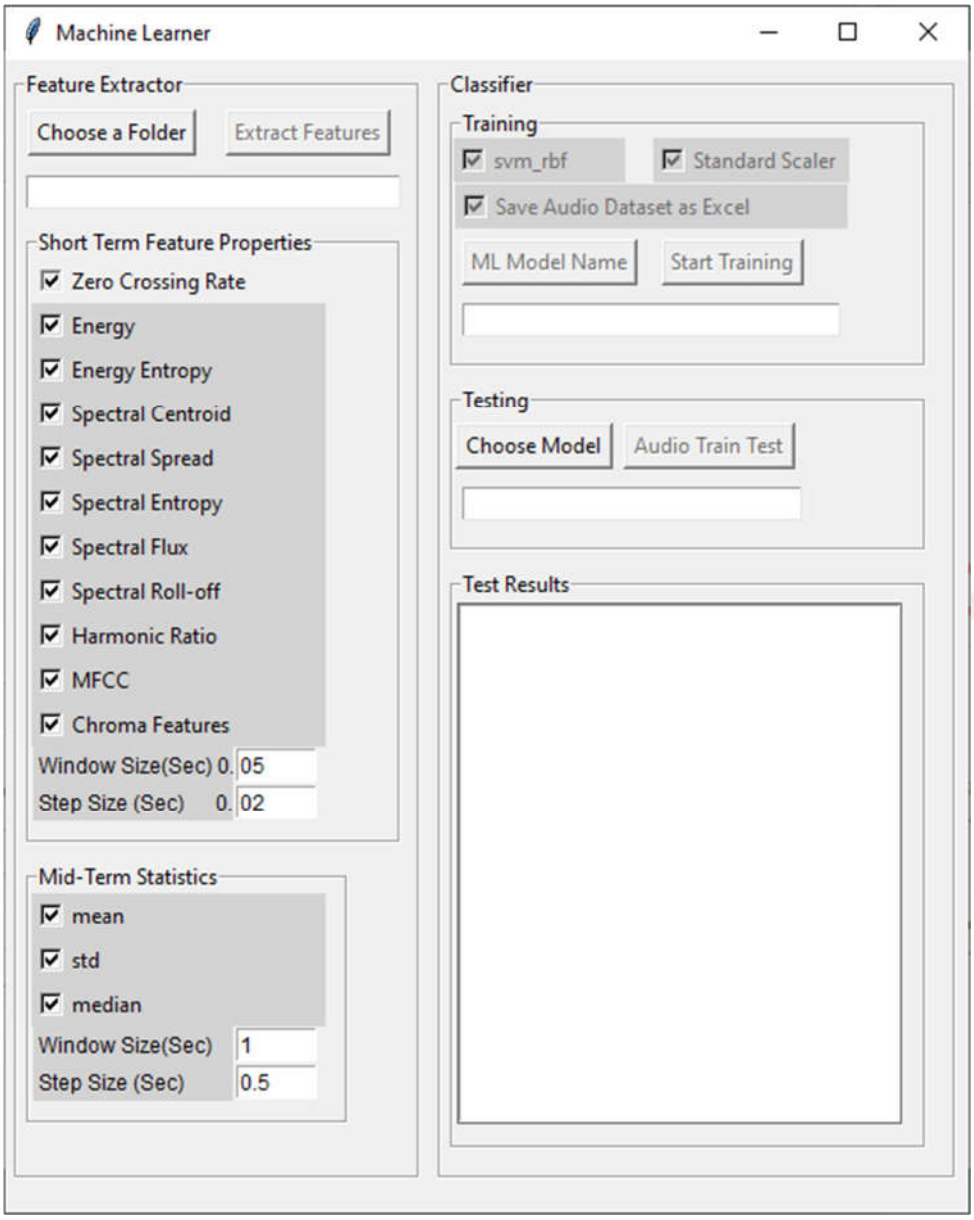

Figure 4.3: MixPrep's machine learner module.

The GUI of the machine learner module was designed as two separate frames: one is Feature Extractor, and the other is Classifier. Even though the pyAudioAnalysis module allows the use of many scikit's machine learning models, for this dissertation

research, only the SVM (RBF) classifier is included in the model training pane because the SVM model with RBF kernel performs better compared to the others.

4.4.1 The Feature Extractor

Default feature extraction preferences are given in Figure 4.3. The user is free to change these properties but at least one short-term besides mid-term features should be selected to instantiate the extraction process. The mid-term window and steps size cannot be shorter than the short-term window and step sizes; if so, the warning message appears.

The *Choose Folder* button, which is responsible for selecting a target folder, brings a folder selection dialogue window in that frame. At that stage, the user must provide a folder in which sub-folders have already been categorized as instrument group names (class names in terms of machine learning terminology). For example, for three instrument group scenarios, the sub-folders might be organized as Drums, Bass and Keyboards. After providing the audio class folder, the Extract Features button turns to an active state, and by clicking on this button, a file dialogue window appears asking the name and path of the audio dataset to save it as an excel file. The feature extraction process starts right after the user has entered the required information and clicked the save button.

After providing the appropriate audio folder, the *Extract Features* button becomes active and by clicking on this button, a file dialogue appears to ask for the audio dataset name and path to save as an excel file. Then, the feature extraction process starts. This feature extraction process can take a long time depending on the number of audio features with their statistics that are being computed, the number of audio files and their lengths in the folders, and computer performance.

Under the *Short-Term Feature Properties* frame, the check box widgets present 11 low-level audio features besides their window-size and step-size calculational durations. Users can create their datasets by different combinations of those audio features with different computational time frames (window and step sizes). Also, the mid-term statistics are restricted with mean, standard deviation, and median. The window and step sizes must be selected for short-term feature extraction between 0.02-0.2 and 0.02-0.01 seconds. On the other hand, those two parameters must be ranged between 0.5-1.5 seconds and 0.1-1 seconds for the mid-term feature extractions.

Whenever a user enters an inappropriate time frame, the software does not initiate the feature extraction process and prompts a warning message where allowed ranges for frame durations are shown on the screen.

4.4.2 The Classifier

The Classifier is necessary for creating and testing a machine learning model for using MixPrep's automatic instrument grouping function. This part has two sections: one for training and the other one for testing ML models. For training, the user must select an appropriate instrument folder by clicking the Choose Folder button from the Feature Extractor frame as the dataset creation process. After providing this condition, the ML Model Name button becomes active, and clicking that button brings a file dialogue box that asks the user to select a machine learning model name and its path for saving it to the hard drive. Next, the user can click the Start Training button to initiate model creation with the provided audio data. However, because the tick box Save Audio Dataset Excel is selected as default, a file dialog appears and expects the user to enter the dataset file name and the saving path into the hard disk as an excel file. The software starts to feature extraction and builds an SVM model by accepting the name and path information.

As seen in the Training frame, the properties, svm_rbf (machine learning model type), Standard Scaler, and Save Audio Dataset as Excel File tick boxes are locked. The model svm_rbf performs best compared to the other available machine learning models in the pyAudioAnalysis library; on the other hand, the standard scalar is a widely used data preparation method for dealing with different ranged parts of data and saving the dataset as an excel file is regarded as beneficial for further investigating the dataset performance on different platforms such as MATLAB. Those properties were intentionally added to the GUI to provide a better understanding for users. For example, considering that a user gives testing as a name, after the training process, four files -testing, *testing.arff*, *testingMeans*, and testing.xlsx- will be saved on the hard drive as a result of the model creation.

This module also has a *Testing* window where users can check the performance of an SVM model with different audio files. To conduct a testing procedure, users should select a machine learning model, one of the ML model files without an extension, and provide a folder containing audio files for testing. The classification results for the

provided audio file(s) (depending on number of the audio files in the testing folder) will be shown in the *Test Results* list box.

4.5 Performance Concern of the Software

Besides higher accuracy, quicker performance is also essential while implementing an instrument classification algorithm into music production steps. The duration of the classification process is substantially affected by the amount and complexity of the data. During the feature extraction process including dataset creation or classification, the duration of the audio files increases the calculation time drastically. On the other hand, audio files in multitrack projects are intact silenced and voiced/played parts get together most of the time. The redundancy of the silence parts becomes problematic, causing increased calculation time and deteriorating the accuracy of the classification.

MixPrep software uses an algorithm that reads the audio files by block-wise technique to shorten the classification duration to attain a more representative version of imported audio files. The algorithm finds the onset point at a higher RMS level of each audio file and saves their two second-duration versions into a temporary file. Then, the execution of the classification process is performed with those two-second excerpts so that the calculation time reduces from a couple of minutes to just seconds.

4.6 Planning the Dataset

The *Machine Learner* is responsible for dataset creation. Because of that, it is worth mentioning some critical points of dataset planning. A dataset is a fundamental part of a machine learning model. The model's success rate depends on the compatibility of the dataset with provided data that will be classified. Therefore, the aim and objective of a machine learning model must be explicitly determined to attain robust learning modeling by a convenient dataset.

In this research, regarding the dataset, the main concerns were to determine the number of instrument classes to have in different folders and organize those folder contents regarding the properties of their audio files. The instrument classes were organized as instrument families like drums, bass, guitars, keyboards, strings, and wind, bearing in mind the general requirements of the mix preparation in this study. Nonetheless, different session preparation ideas considering multitrack mixing styles or strategies

applied by the engineers may require different solutions, and they are rather more challenging to implement as an automatic solution like this study aims to achieve.

Another point is that, finding or creating a decent audio library for building an audio dataset is not an easy task. Deciding which audio files are included in the audio library, organizing them, and editing requirements of the audio files (like taking out the unwanted part) are time-consuming tasks. A detailed dataset creation approach will be discussed in the experiment section.

5. EXPERIMENT WITH THE MIXPREP

This section demonstrates a complete procedure that includes obtaining an audio library, creating an ML model related to the audio dataset, and estimating the MixPrep software performs with real multitrack projects.

In order to persuade this purpose, suitable audio files are investigated in the Apple Jam Packs library for dataset creation. This library content is organized in six folders regarding instrument family classes. Table 5.1 shows the properties of these six folders. The next step was to extract audio features to create the audio dataset and its ML model. The settings of the window and step size durations were 20 msec and 40 msec for short-term feature extraction, and for mid-term feature extraction, these parameters were 400 and 800 msec respectively. The extraction process included 35 audio features (Table 5.2) and three different statistics (mean, standard deviation by mean, and median), resulting in a 5276x106 (included the target instrument family) dimensional dataset and an SVM (with RBF kernel). The model evaluation showed that the SVM model performs with 95.0% accuracy with parameter C equals 20. The same dataset was also reviewed by using MATLAB's classification learner module. As a result, the SVM Cubic algorithm attained 95.1% accuracy among the 24 different machine learning algorithms (Table 5.3). This result proves that the SVM algorithm is the best machine learning algorithm for this type of audio application. Additionally, 95.0% classification accuracy shows that the organization of the instrument classes is valid.

This ML model is used to estimate instrument family of 1423 audio files in multitrack projects from *Mixing Secrets for The Small Studio-Additional Resources* website[7]. Four musical genres (pop, rock, jazz, and dance) are obtained, and 20 multitrack projects are picked from each on that website.

[7] Mixing Secrets For The Small Studio - Additional Resources

Among the Pop, Rock, Jazz, and Electronic/Dance genres, the average performance results attained by the machine learning model are 65.89%, 68.52%, 67.20%, and 57.10% respectively. On the other hand, the average performance results for dedicated instrument families that the Drums& Percussion, Bass, Guitars, Keyboards, Strings, and Wind were 84.80%, 67.72%, 53.66%, 16.44%, 61.11%, and 25.0%, respectively. Table 5.15 shows the success rates for instrument families and genres. Likewise, the selected multitrack projects and the estimation results of each audio file were given in Appendix B and C.

5.1 The Experiment Dataset

According to the aim of this dissertation, an audio dataset is necessary for an instrument's family recognition. For that purpose, the dataset must comprise a wide range of audio files exhibiting properties of specific instrument family groups. In music production applications in which the machine learning methods take part, the audio loops can be more functional than the general-purpose audio libraries. Therefore, Apple Jampack Loops were used to create this dissertation dataset.

During the dissertation research, the researcher examined the contents of Jam Pack 1, Jam Pack Remix Tools, Jam Pack Rhythm Section, Jam Pack Symphony Orchestra. Even though the selected audio files had different lengths, no further editing seemed necessary. The audio files in those loop libraries were arranged into six instrument families named Drums & Percussions, Bass, Guitars, Keyboards, Strings, and Winds to create an audio dataset. The properties of the instrument classes and the number of audio files in each are available in Table 5.1. Although properties of the audio contents and the number of instrument classes differ on the purpose of the classification task, this setup is considered appropriate for this dissertation research.

For the research dataset, the audio features of the MixPrep machine learner are given in Table 5.2. The dataset consists of 11 temporal and spectral features with means, standard deviations, and median statistics. The resulting audio dataset has dimensions of 5276x106 (the latest column for corresponded class names). For this dataset, window size of short-term calculations is 20 msec, and the step size is 40 msec; for the mid-term, parameters are 400 msec and 800 msec, respectively.

Table 5.1: Instrument families (classes) in seven instrument groups and their properties.

Classes	Some Properties of the Audio Classes	Min. Max. Length	Number of Audio Files
Drums & Percussions	Drum-sets (Patterns and fills, in different styles), Electronic-Dance Drums, Drum-set with percussion(s), Drums Machines, Percussions (such as congas, Tambourine, Shakers and similar.)	1-20 sec.	1965
Bass	Bass guitars (in different genres, with some, availability of audio effects, some sampled), Double basses (played with different techniques)	1-23 sec.	843
Guitars	Wide variety of Guitars such as Acoustic, Nylon, Electric, with various genres (rock, metal, jazz and similar) and playing techniques (fingers, plucked, strummed, slides, and similar), most of them having some audio some effects (like distortion, wah-wah, reverb)	1-32 sec.	1114
Keyboards	Organs, Clavinets, Wurlitzers, Rhodes, Acoustic and Electronic Pianos.	1-32 sec.	599
Strings	Mostly orchestral, sample-based, or real. Various playing techniques, some recordings solo, but most of them recorded as an ensemble. Reverb as the predominant effect.	1-32 sec.	395
Winds	Brasses, Harmonica, Flutes, French and English Horns, Clarinets, Oboes. Some solo performances but most of them as an ensemble.	1-40 sec.	359

Table 5.2: Audio features and number of columns.

Features	Number of Columns
Zero-Crossing Rate	Single
Energy (Power)	Single
Entropy of Energy	Single
Spectral Centroid	Single
Spectral Spread	Single
Spectral Entropy	Single
Spectral Flux	Single
Spectral Roll-off	Single
Harmonic Ratio	Single
MFCC	13
Chroma Vector	12+1 (mean of the bins)

5.2 Estimation of the Dataset Consistency with Supervised Machine Learning Algorithms

The MixPrep provides two tables while creating the audio dataset. The first one shows the best C (regularization) parameter, providing each class name with PRE, REC, and f1 merits for the SVM algorithm. The second one shows a confusion matrix for the best C parameter that performs best among the others. These tables are given in the Appendix A. According to the evaluation result, the best C parameter is 20.0, which gives 94.2% accuracy and 92.0% f1 for the dataset.

The same dataset was also reviewed with MATLAB's ClassificationLearner module for double-checking while finding the best-supervised machine learning method. According to the result, the best audio classification method was the SVM Cubic algorithm with 94.3% overall accuracy with five folded cross-validation. Table 5.3 shows the supervised machine learning algorithms with various model/kernel types and their accuracy percentages for the provided audio dataset.

Table 5.3: Accuracy results by MATLAB's ClassificationLearner module for the dataset.

Classification Algorithm	Model Type/Kernel	Overall Accuracy
Tree	Fine	80,5 %
	Medium	75,4 %
	Coarse	70,4 %
Discriminant	Linear	83,5 %
	Quadratic	85,7 %
Bayesian	Naive	71,6 %
	Kernel Naive	75,0 %
SVM	Linear	87,0%
	Quadratic	93,3 %
	Cubic	94,3 %
	Fine Gaussian	52,8 %
	Medium Gaussian	91,4 %
	Course Gaussian	80,9 %
KNN	Fine	89,0 %
	Medium	84,4 %
	Coarse	75,4 %
	Cosine	84,2 %
	Cubic	82,2 %
	Weighted	87,1 %
Ensemble	Boosted Trees	79,3 %
	Bagged Trees	88,9 %
	Subspace Discriminant	81,8 %
	Subspace KNN	85,2 %
	RUSBoosted Trees	73,6 %

For the SVM Cubic algorithm, confusion matrixes of Numbers of Observations for each class, TPR-FNR (True Positive Rates-False Negative Rates) and PPV-FDR (Positive Predictive Values- False Discovery Rates) were given in Table 5.4, Table 5.5, and Table 5.6, respectively.

In Table 5.4, the diagonally placed cells in which the True and the Predicted classes coincide, showing the number of successful observations for each instrument class by the provided dataset.

Table 5.4: Numbers of observations for each class in the provided dataset.

True Class \ Predicted Class	Bass	Drums & Percussions	Guitar	Keyboards	Strings	Wind
Bass	777	23	32	6	5	0
Drums & Percussions	21	1793	99	31	11	10
Guitar	18	46	875	84	62	29
Keyboards	4	10	129	394		37
Strings	13	16	139	18	199	10
Wind	2	14	56	56	21	210

The divisions of the successful observation results (in Table 5.4) to the number of audio files in individual classes provide the True Positive Rates for each of these instrument classes, given in Table 5.5.

Table 5.5: Confusion matrix TPR (True Positive Rates) and FNR (False Negative Rate).

True Class \ Predicted Class	Bass	Drums & Percussions	Guitar	Keyboards	Strings	Wind	TPR	FNR
Bass	92.2%	2.7%	3.8%	0.7%	0.6%	0.0%	92.2%	7.8%
Drums & Percussions	1.1%	91.2%	5.0%	1.6%	0.6%	0.5%	91.2%	8.8%
Guitar	1.6%	4.1%	78.5 %	7.5%	5.6%	2.6%	78.5%	21.5%
Keyboards	0.7%	1.7%	21.5 %	65.8 %	4.2%	6.2%	65.8%	34.2%
Strings	3.3%	4.1%	35.2 %	4.6%	50.4 %	2.5%	50.4%	49.6%
Wind	0.6%	3.9%	15.6 %	15.6 %	5.8%	58.5 %	58.5%	41.5%

For this dataset, considering the confusion matrix of TPR and FNR's (in Table 5.5), the prediction performance is the best for the Bass class (97.3%), while the String class is the worst. However, according to the confusion matrix for PPV and FPV (in Table 5.6), the classification performance for Drums&Percussion class is best (98.3%); yet for the Wind class, it is the worst (74.8%).

For each instrument class, the classification results were sorted from the best to the worst as Bass, Drums&Percussions, Guitars, Keyboards, Wind, and Strings in terms of TPR and FNR merits (Table 5.5). However, according to the PPV and FDR confusion matrix (Table 5.6), the performance results from best to worst are Drums&Percussions, Bass, Strings, Guitars, Keyboards, and Winds.

The best accuracy rate attained with the SVM Cubic Kernel endorses the idea that MixPrep's SVM method is a suitable machine learning method for this audio dataset.

Table 5.6: Confusion matrix for PPV (Positive Predictive Values) and FDR (False Discovery Rates).

True Classes	Bass	Drums & Percussions	Guitar	Keyboards	Strings	Wind
Bass	93.1	1.2	2.4	1.0	1.5	
Drums & Percussions	2.5	94.3	7.4	5.3	3.4	3.4
Guitar	2.2	2.4	65.8	14.3	19.2	9.8
Keyboards	0.5	0.5	9.7	66.9	7.7	12.5
Strings	1.6	0.8	10.5	3.1	61.6	3.4
Wind	0.2	0.7	4.2	9.5	6.5	70.9
PPV	93.1	94.3	65.8	66.9	61.6	70.9
FDR	6.9	5.7	34.2	33.1	38.4	29.1

Predicted Classes

5.3 The Experiment with Multitrack Projects

MixPrep's instrument classification performance was tested with 80 multitrack projects taken from Mixing Secrets' Free Multitrack Download Library[8]. The projects were picked considering four different musical genres grouped as Pop, Rock, Jazz, Electronic/Dance and imported into the software one by one to simulate in real-life music production situations. Some of those projects contain vocals and back-vocal tracks. However, vocal-related tracks were not included in the statistical calculation of the classification success because the dissertation dataset does not have any dedicated vocal class. On the contrary, the tracks with "FX" tags were included in the calculation of the classification success since those tracks generally have rhythmic functions rather

[8] https://www.cambridge-mt.com/ms/mtk/

than melodic; therefore, they are expected to be categorized in Drums&Percussion class.

It is necessary to remark that, for estimating a success rate, the number of results in each instrument family estimated by the instrument recognition algorithm was divided by the sum of the instrument. The number of each instrument family was determined only regarding the file names. So, the computational merit for success rates depends on the description of the audio files, even though it sometimes contradicts their file contents or arrangement-wise functioning. For example, an audio file named bass may be created with an electronic piano, or a file named synth may remind a strings instrument.

5.3.1 Multitracks under the Pop genre

There are 384 audio files examined for this genre, 253 of which were found correctly according to the instrument family classification (Table 5.7).

Table 5.7: MixPrep's machine learner performance evaluations for the Pop projects.

Genre		Artist/Band	Song Name	Numbers of Successful Tracks	Numbers of Tracks	Instrument Classification Success
Pop	1	Atlantis Bound	It Was My Fault for Waiting	20	21	**95,24%**
	2	David Tyo	Oh Life	15	17	**88,24%**
	3	David Tyo	Long Way Home	13	15	**86,67%**
	4	Amy Helm & The Handsome Strangers	Rescue Me	12	15	**80,00%**
	5	Benjamin John	Better Way	18	23	**78,26%**
	6	The Wrong'uns	Rothko	8	11	**72,73%**
	7	Ben Carrigan	Well Talk About It All Tonight	20	29	**68,97%**
	8	Finlay	Same Kind of Life	9	14	**64,29%**
	9	BaumXmedia	Koishii	12	19	**63,16%**
	10	David Tyo	Never Ebb But Flow	15	24	**62,50%**
	11	Eduard Semenov	River of The White Gloom	5	8	**62,50%**
	12	Ted Behrman	Convertible	11	18	**61,11%**
	13	David Tyo	Its So Easy to Love You	9	15	**60,00%**
	14	Angelo Boltini	This Town	26	43	**60,47%**
	15	Drumtracks	Ghost Bitch	8	14	**57,14%**
	16	Al James	Schoolboy Fascination	13	23	**56,52%**
	17	Trick Bird	Window	10	17	**58,82%**
	18	Ben Carrigan	Hey Carrie Anne	10	19	**52,63%**
	19	Strobe	Maya	11	22	**50,00%**
	20	James Elder & Mark M Thompson	The English Actor (Full)	8	17	**47,06%**
		Number of successful and total tracks for this genre		253	384	
			Average Success			**65,89%**

The best family identification result was achieved at a 95.24% level, resulting in 20 correct classifications over the 21 audio files for the project called *It Was My Fault for Waiting*. This project consists of Drum&percussion, Bass, and Guitars families, and the common sonic feature among the tracks is that they all have a very dry and clean sound.

In Table 5.8, classification success rates for each track are given for the available instrument families. Through the tracks, the average success rates for the Drums&percussion, Bass, Guitars, Keyboards, Strings, and Wind families are 83.78%, 76.00%, 53.33%, 22.92%, 60,61%, 0.0%, respectively.

Table 5.8: Pop projects by the success rates of the available instrument families.

	Artist/Band	Track Name	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
1	Atlantis Bound	It Was My Fault for Waiting	100,00%	100,00%	91,67%	-	-	-
2	David Tyo	Oh Life	90,00%	100,00%	75,00%	100,00%	100,00%	-
3	David Tyo	Long Way Home	100,00%	100,00%	80,00%	50,00%	-	-
4	Amy Helm & The Handsome Strangers	Rescue Me	100,00%	100,00%	33,33%	-	0,00%	-
5	Benjamin John	Better Way	93,75%	100,00%	40,00%	0,00%	-	-
6	The Wrong'uns	Rothko	80,00%	100,00%	60,00%	-	-	-
7	Ben Carrigan	Well Talk About It All Tonight	78,57%	100,00%	100,00%	33,33%	66,67%	0,00%
8	Finlay	Same Kind of Life	57,14%	100,00%	100,00%	33,33%	-	-
9	BaumXmedia	Koishii	100,00%	100,00%	-	22,22%	100,00%	-
10	David Tyo	Never Ebb But Flow	81,82%	100,00%	50,00%	0,00%	-	-
11	Eduard Semenov	River of The White Gloom	100,00%	100,00%	50,00%	-	-	0,00%
12	Ted Behrman	Convertible	100,00%	50,00%	33,33%	0,00%	-	-
13	David Tyo	Its So Easy to Love You	85,71%	-	50,00%	0,00%	0,00%	-
14	Angelo Boltini	This Town	93,33%	100,00%	41,67%	0,00%	62,50%	-
15	Drumtracks	Ghost Bitch	50,00%	0,00%	-	100,00%	100,00%	-
16	Al James	Schoolboy Fascination	66,67%	100,00%	25,00%	0,00%	100,00%	-
17	Trick Bird	Window	77,78%	0,00%	50,00%	-	-	-
18	Ben Carrigan	Hey Carrie Anne	66,67%	100,00%	0,00%	100,00%	37,50%	-
19	Strobe	Maya	90,91%	50,00%	0,00%	0,00%	-	-
20	James Elder & Mark M Thompson	The English Actor (Full)	71,43%	0,00%	33,33%	33,33%	-	-
		Average	**83,78%**	**76,00%**	**53,33%**	**22,92%**	**60,61%**	**0,00%**

5.3.2 Multitracks under the Rock genre

In this section, the algorithm is correctly classified 259 audio tracks to instrument families by examining 378 audio tracks in total (Table 5.9). There are no strings and wind instruments available in the multitrack projects of this genre. The song *The Island of Alsocanla* was scored with a complete (100.0%) rate by the successful placement of 12 instruments under the Drums&percussion, Bass, and Guitars families (Table 5.10).

Table 5.9: MixPreps's machine learner performance evaluations for the Rock projects.

Genre		Artist/Band	Song Name	Numbers of Successful Tracks	Numbers of Tracks	Instrument Classification Success
Rock	1	Imp Act	The Island of Alsocanla	12	12	**100,00%**
	2	Blue Lit Moon	Dad's Glad	9	10	**90,00%**
	3	Candlebox	Happy Pills	13	15	**86,67%**
	4	Complainiacs	Etc	9	11	**81,82%**
	5	Decypher	Unseen	22	27	**81,48%**
	6	Bobby Nobody	Stitch Up	10	13	**76,92%**
	7	Forkupines	Sleep by The Fire Bloom in Water	23	30	**76,67%**
	8	The Black Crown	Flames	15	20	**75,00%**
	9	All Hands Lost	Ambitions	13	18	**72,22%**
	10	Angels In Amplifiers	I'm Alright	7	10	**70,00%**
	11	Candlebox	Surrendering	13	19	**68,42%**
	12	Louis Cressy Band	Good Time	13	19	**68,42%**
	13	Bill Chudziak	Children Of No-one	14	21	**66,67%**
	14	The Apprehended	Still Flyin	16	25	**64,00%**
	15	Ale Lak	Nosso Mundo Deixou De Existir	15	24	**62,50%**
	16	Cnoc An Tursa	Bannockburn	18	31	**58,06%**
	17	The Black Crown	Cage	13	24	**54,17%**
	18	Death of a Romantic	The Well	10	19	**52,63%**
	19	Jet B	Suit You	7	14	**50,00%**
	20	DarkRide	Dead Enemies	7	16	**43,75%**
		Number of successful and total tracks for this genre		259	378	
			Average Success			**68,52%**

Table 5.10: Rock projects by the success rates of the available instrument families.

	Artist/Band	Track Name	Drums & Percussions	Bass	Guitars	Keyboards
1	Imp Act	The Island of Alsocanla	100,00%	100,00%	100,00%	-
2	Blue Lit Moon	Dad's Glad	85,71%	100,00%	100,00%	-
3	Candlebox	Happy Pills	100,00%	66,67%	50,00%	-
4	Complainiacs	Etc	71,43%	100,00%	100,00%	-
5	Decypher	Unseen	100,00%	100,00%	58,33%	-
6	Bobby Nobody	Stitch Up	100,00%	100,00%	25,00%	-
7	Forkupines	Sleep by The Fire Bloom in Water	90,48%	50,00%	50,00%	0,00%
8	The Black Crown	Flames	84,62%	50,00%	60,00%	-
9	All Hands Lost	Ambitions	91,67%	100,00%	0,00%	-
10	Angels In Amplifiers	I'm Alright	60,00%	100,00%	100,00%	0,00%
11	Candlebox	Surrendering	80,00%	100,00%	33,33%	-
12	Louis Cressy Band	Good Time	64,29%	100,00%	100,00%	0,00%
13	Bill Chudziak	Children Of No-one	100,00%	0,00%	60,00%	20,00%
14	The Apprehended	Still Flyin	100,00%	100,00%	60,00%	0,00%
15	Ale Lak	Nosso Mundo Deixou De Existir	83,33%	100,00%	52,94%	-
16	Cnoc An Tursa	Bannockburn	81,25%	0,00%	41,67%	0,00%
17	The Black Crown	Cage	63,64%	66,67%	44,44%	0,00%
18	Death of a Romantic	The Well	62,50%	50,00%	50,00%	0,00%
19	Jet B	Suit You	66,67%	100,00%	0,00%	0,00%
20	DarkRide	Dead Enemies	60,00%	100,00%	0,00%	-
		Average	**82,50%**	**79,49%**	**51,24%**	**5,56%**

5.3.3 Multitracks under the Jazz genre

This part comprises 314 individual audio tracks in which the ML model was able to classify 211 of them correctly, which corresponds to 67.20%. In this part, even though each instrument family contains audio files, again the Drums&percussion is the most crowded class (176 tracks); yet, the strings have only a few (3 tracks). As seen in Table 5.11, the song *If I Were A Bell* has the highest instrument classification score at 90.0%, and the song *The Opener* has the lowest classification score at 21.05%, which is also the poorest result among all the genres.

Table 5.11: MixPrep's machine learner performance evaluations for the Jazz projects.

Genre		Artist/Band	Song Name	Numbers of Successful Tracks	Numbers of Tracks	Instrument Classification Succes
Jazz	1	H-owl Project	If I Were A Bell	9	10	**90,00%**
	2	Lorenzo Price	Changing Things	14	16	**87,50%**
	3	Patrick Talbot	Upper Hand	17	20	**85,00%**
	4	Jesper Buhl Trio	What Is This Thing Called Love	5	6	**83,33%**
	5	Dino On The Loose	Queens Light	14	17	**82,35%**
	6	Patrick Talbot	Blue	12	15	**80,00%**
	7	Patrick Talbot	A Reason to Leave	19	24	**79,17%**
	8	Patrick Talbot	Fool	10	13	**76,92%**
	9	Maurizio Pagnutti Sextet	Bess	12	16	**75,00%**
	10	Maurizio Pagnutti Sextet	All The Gin Is Gone (full)	12	16	**75,00%**
	11	Araujo	The Saga of Harrison Crabfeathers	8	11	**72,73%**
	12	Alejo Granados	Rumba Chonta	13	18	**72,22%**
	13	Eddie Garrido	Africa	10	14	**71,43%**
	14	Eddie Garrido	Scarlett	16	23	**69,57%**
	15	Patrick Talbot	Set Me Free	10	16	**62,50%**
	16	Eddie Garrido	5th Floor	5	10	**50,00%**
	17	Pawel Maciwoda	1 Of The First	13	30	**43,33%**
	18	Abletones Big Band	Song of India	7	20	**35,00%**
	20	Abletones Big Band	Corine Corine	5	19	**26,32%**
	19	Lindy Hip Big Band	The Opener	4	19	**21,05%**
		Number of successful and total tracks for this genre		211	314	
			Average Success			**67,20%**

Table 5.12 shows the success rates of every instrument family for each available track in this genre. Still, the average classification performance is the best for the Drums&percussion family with 85.23%, and the other scores for Bass, Guitars, Keyboards, Strings, and Wind are 64.29%, 62,07%, 32.43%, 66.67%, 26.83%, respectively.

Table 5.12: Jazz projects by the success rates of the available instrument families.

	Artist/Band	Track Name	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
1	H-owl Project	If I Were A Bell	100,00%	100,00%	-	0,00%	-	-
2	Lorenzo Price	Changing Things	100,00%	100,00%	50,00%	0,00%	-	-
3	Patrick Talbot	Upper Hand	100,00%	100,00%	100,00%	0,00%	-	0,00%
4	Jesper Buhl Trio	What Is This Thing Called Love	100,00%	0,00%	-	100,00%	-	-
5	Dino On The Loose	Queens Light	100,00%	100,00%	-	50,00%	-	-
6	Patrick Talbot	Blue	81,82%	100,00%	100,00%	0,00%	-	-
7	Patrick Talbot	A Reason to Leave	82,35%	0,00%	75,00%	100,00%	-	-
8	Patrick Talbot	Fool	75,00%	100,00%	66,67%	100,00%	-	-
9	Maurizio Pagnutti Sextet	Bess	100,00%	50,00%	-	0,00%	-	66,67%
10	Maurizio Pagnutti Sextet	All The Gin Is Gone (full)	100,00%	100,00%	-	0,00%	-	33,33%
11	Araujo	The Saga of Harrison Crabfeathers	71,43%	100,00%	100,00%	0,00%	-	-
12	Alejo Granados	Rumba Chonta	88,89%	50,00%	100,00%	-	-	50,00%
13	Eddie Garrido	Africa	100,00%	100,00%	50,00%	50,00%	0,00%	66,67%
14	Eddie Garrido	Scarlett	75,00%	100,00%	100,00%	20,00%	100,00%	-
15	Patrick Talbot	Set Me Free	80,00%	0,00%	33,33%	100,00%	-	0,00%
16	Eddie Garrido	5th Floor	75,00%	100,00%	0,00%	25,00%	-	-
17	Pawel Maciwoda	1 Of The First	81,82%	33,33%	33,33%	0,00%	-	16,67%
18	Abletones Big Band	Song of India	44,44%	50,00%	0,00%	50,00%	-	16,67%
20	Abletones Big Band	Corine Corine	62,50%	0,00%	0,00%	0,00%	-	0,00%
19	Lindy Hip Big Band	The Opener	100,00%	100,00%	0,00%	0,00%	-	0,00%
		Averages	**85,23%**	**64,29%**	**62,07%**	**32,43%**	**66,67%**	**26,83%**

5.3.4 Multitracks under the Electronic/Dance genre

In this part, the multitrack projects contain 352 audio files, and the ML algorithm was able to identify the instrument families for 201 of them, which coincide with 57.10%. As seen in Table 5.13, the project *Horizon* gained a higher rank at 76.92% among the other projects in this genre, while *Fly High* attained the worst score (39.13%). Those projects do not have any audio track that belongs to the Strings or Wind families.

Table 5.14 shows the success rates of every instrument family for each available track in this genre. Again, the average classification performance is the best for the Drums&percussion family with 87.83%, besides Bass, Guitars, keyboards have 51.43%, 66.67 and 10.66%, respectively. So, it is observed that the Keyboards family cannot represent synthesizer-based audio files that dominate this genre.

Table 5.13: Mixprep's machine learner performance evaluation for the Electronic/Dance projects.

Genre		Artist/Band	Song Name	Numbers of Successful Tracks	Numbers of Tracks	Instrument Classification Succes
Electronic/Dance	1	Galias	Horizon	10	13	**76,92%**
	2	AM Contra	Heart Peripheral	6	8	**75,00%**
	3	Albert Kader	Ubiquitous	8	11	**72,73%**
	4	Mustafa Albazy	Shela Ya Marhaba	25	35	**71,43%**
	5	Babe Grand	Aiguille Rouge	16	23	**69,57%**
	6	c:fx	Mathematician	11	16	**68,75%**
	7	Babe Grand	King Of The Weekend	10	15	**66,67%**
	8	APZX	Transcention	10	16	**62,50%**
	9	Albert Kader	Whiptails	8	13	**61,54%**
	10	Ian Dearden	Terania Creek Walking	7	12	**58,33%**
	11	J0K3R	SeptemberTrance	10	18	**55,56%**
	12	Mustafa Albazy	Tgheer	17	32	**53,13%**
	13	APZX	Lacuna	13	25	**52,00%**
	14	Skelpolu	Anomalous Weeping (Full)	6	12	**50,00%**
	15	JP Lantieri	Chardonnay Mystik Vibe Remix	6	13	**46,15%**
	16	Skelpolu	Entwine	6	13	**46,15%**
	17	JP Lantieri	Chardonnay	5	12	**41,67%**
	18	Nervbloc	Slapback	12	27	**44,44%**
	19	Skelpolu	Cold Strive	6	15	**40,00%**
	20	Aron Jaeger	Fly High	9	23	**39,13%**
		Number of successful and total tracks for this genre		201	352	
			Average Success			**57,10%**

Table 5.14: Electronic/Dance projects by the success rates of the available instrument families.

	Artist/Band	Track Name	Drums & Percussions	Bass	Guitars	Keyboards
1	Galias	Horizon	100,00%	100,00%	-	0,00%
2	AM Contra	Heart Peripheral	100,00%	100,00%	-	0,00%
3	Albert Kader	Ubiquitous	100,00%	100,00%	-	0,00%
4	Mustafa Albazy	Shela Ya Marhaba	80,95%	100,00%	60,00%	50,00%
5	Babe Grand	Aiguille Rouge	77,78%	100,00%	-	25,00%
6	c:fx	Mathematician	87,50%	100,00%	-	33,33%
7	Babe Grand	King Of The Weekend	75,00%	100,00%	-	0,00%
8	APZX	Transcention	83,33%	0,00%	-	0,00%
9	Albert Kader	Whiptails	100,00%	0,00%	-	0,00%
10	Ian Dearden	Terania Creek Walking	75,00%	0,00%	-	33,33%
11	J0K3R	SeptemberTrance	75,00%	100,00%	-	25,00%
12	Mustafa Albazy	Tgheer	100,00%	50,00%	-	6,67%
13	APZX	Lacuna	100,00%	50,00%	-	0,00%
14	Skelpolu	Anomalous Weeping (Full)	100,00%	0,00%	100,00%	0,00%
15	JP Lantieri	Chardonnay Mystik Vibe Remix	100,00%	0,00%	-	0,00%
16	Skelpolu	Entwine	100,00%	0,00%	-	28,57%
17	JP Lantieri	Chardonnay	66,67%	50,00%	-	0,00%
18	Nervbloc	Slapback	90,91%	28,57%	-	0,00%
19	Skelpolu	Cold Strive	100,00%	50,00%	-	0,00%
20	Aron Jaeger	Fly High	80,00%	100,00%	-	0,00%
		Averages	**87,83%**	**51,43%**	**66,67%**	**10,66%**

5.4 Evaluation of The Results

Table 5.15 gives the relationship between instrument families and genres, providing individual and overall success rates. The number of examined audio tracks in each genre is almost equal, gathered around 350. The ML model successfully placed 924 audio files among 1428 files to the related instrument families, corresponding to a 64.71% success rate.

Table 5.15: Success rates for the instrument families and genres.

			Genres				
			Pop	**Rock**	**Jazz**	**Electronic/Dance**	**Tracks**
Instrument Families	**Drums & Percussion**	correct	155	165	150	166	636
		total	185	200	176	189	750
		success rates	**83,78%**	**82,50%**	**85,23%**	**87,83%**	**84,80%**
	Bass	correct	19	31	18	18	86
		total	25	39	28	35	127
		success rates	**76,00%**	**79,49%**	**64,29%**	**51,43%**	**67,72%**
	Guitars	correct	48	62	18	4	132
		total	90	121	29	6	246
		success rates	**53,33%**	**51,24%**	**62,07%**	**66,67%**	**53,66%**
	Keyboards	correct	11	1	12	13	37
		total	48	18	37	122	225
		success rates	**22,92%**	**5,56%**	**32,43%**	**10,66%**	**16,44%**
	Strings	correct	20	-	2	-	22
		total	33	-	3	-	36
		success rates	**60,61%**		**66,67%**		**61,11%**
	Wind	correct	0	-	11	-	11
		total	3	-	41	-	44
		success rates	**0,00%**		**26,83%**		**25,00%**
		correct	253	259	211	201	924
		total	384	378	314	352	1428
		Averages for Genres	**65,89%**	**68,52%**	**67,20%**	**57,10%**	**64,71%**

Regarding instrument families, a generous amount of rhythmic instrument tracks exists inside the 80 multitrack projects (750 files), and with the provided dataset, the ML model concluded that 636 of them belong to the Drums&Percussions class, bringing the best classification score as 84.80%. The available guitar collection which has the second-largest instrument population in this experiment contains 246 tracks, 132 of which are estimated by the algorithm successfully at a 53.66% rate. With 222 audio

files, the keyboards compose the third largest collection among the entire audio files. Unfortunately, the algorithm determined 37 of those files correctly, which is the worst classification performance among instrument families' success rate, which is 16.44%. On the other hand, bass tracks constitute the fourth largest instrument collection by containing 127 tracks. The algorithm associated 86 of those correctly at a 67.72% success rate which is the second-best performance result among the instrument families. The wind instrument collection is the fifth largest collection, comprised of 44 audio tracks and the algorithm successfully classified only 11 of them at a 25.0% rate, which is the second worst classification performance in the whole experiment. The Strings is the smallest collection containing only 36 instruments; yet, the algorithm estimated only 22 audio files at a 61.11% rate.

The distribution of the success rates for each instrument group is also worth mentioning regarding their placement under individual genres. As shown in Table 5.15, the success rates are similar throughout the Drums&Percussion family, from 82.50% to 87.83% for all genres. This result indicates that the ML model and dataset perform quite successfully for the classification of the rhythmic instruments.

For the Bass family classification, the best success rate was 79.49% for the Rock genre, and the worst result was 51.43% for the Electronic/Dance genres. The ML did wrong family estimations for bass files in *Transcention*, *Whiptails*, *Terania Creek Walking*, *Anomalous Weeping*, *Chardonnay Mystik Vibe Remix*, *Entwine*, and *Slapback* projects. When listening to the bass-tagged audio files from those seven projects, it has been observed that some audio files do not sound like a regular bass guitar. Also, some of them were processed a modulated low-pass filter (*Terania Creek Walking*); triggered with distinct rhythmic arpeggios (*Whiptails*); some comprised of a layered sound (in *Anomalous Weeping and Entwine*); some had a very short attack and sustained parts that reminded of a kick drum (Chardonnay Mystik Vibe Remix), or a combination of these mentioned properties though probable. Likewise, in some situations, the ML algorithm did some reasonable estimations while examining the instrument family of a track, even if its actual name dictates a different instrument. One good example of that situation came out while reviewing 11_Bass3b.wav, 13_Bass4.wav, 14_Bass5.wav tracks from the *Slapback* project. These audio files sound pretty similar to a distortion guitar, which explains why the algorithm recognized them as Guitars.

The success rates of guitar classification through the genres in Table 5.15 indicate that the Rock genre attained a lower rate (51.24%), even though this instrument type is fundamental for this specific genre. In order to understand why almost half of those tracks were classified wrongly, all guitar tracks from the worst scored project called *Dead Enemies* were investigated. The tracks were divided manually regarding their onsets, and 61 audio excerpts were acquired. Then, these excerpts were imported to the MixPrep as if a project were exposed to a new instrument family classification procedure. Although the ML algorithm had previously considered the actual guitar files as a member of the Drums&percussion family, this time, 72.13% correct classification rate occurred. So, the result shows that a robust audio excerpt algorithm is required to attain more accurate instrument recognition results. The excerpt numbers for each guitar track from the project *Dead Enemies* and their number of successful classifications and rates are shown in Table 5.16.

Table 5.16: The name of the guitar tracks and their excerpts.

	12_ElecGtr01	13_ElecGtr01DT	14_ElecGtr02	15_ElecGtr03	16_ElecGtr04	Total
Correct Results	13	12	7	6	6	44
Number of Excerpts	13	12	18	9	9	61
Success	100%	100%	38.89%	66.67%	66.67%	72.13%

The Keyboard instrument family estimation showed the worst result compared to success rates of other family classes. In this study, the synthesizer-based audio files were assumed as a member of the Keyboard family. This assumption creates uncertainty in the Keyboard instrument family definition in the dataset because synthesized sounds have unlimited boundaries in terms of timbre that sometimes, however, can be associated with a wide variety of instruments. Thus, this assumption has deteriorated the calculation of the success rate for this specific instrument family. For example, in the *Still Flyin* project (from the Rock genre), the algorithm classified the 22_Synth2.wav track as a part of the Drums&percussion class; nevertheless, this track has a noticeable string sound. After splitting this audio file into three pieces regarding its sounding parts and then classifying them again, this time, the ML algorithm was able to estimate two pieces as a member of the String family. Additionally, despite 21_Synth1.wav and 23_Synth3.wav having the phrase "synth" in their file names, they sound like

a string, which the algorithm had also guessed. Therefore, the sound ambiguity (name and function conflict) and onset detection irregularity are apparent aspects of designing an ML-based instrument recognition implementation in the mix-preparation.

As seen in Table 5.15, only two genres have string-tagged audio files through the experiment's multitrack projects. The Pop accommodates 33, and the Jazz holds only three string-related audio files. The project *Hey Carrie Anne* (Pop) has eight-string tracks, only three of which were classified in the Strings family. When analyzing misclassified tracks in this project, 11_Cello.wav, 09_Violin2.wav, and 14_ViolaSamples.wav, it was observed that those audio files are polyphonic multi-instrument recordings, which are out of the scope of this dissertation. So, these audio files are out of context. At the same time, even though 13_ViolinSamples.wav and 15_CelloSamples.wav tracks are monophonic performances, classification improvement is only attained by the second one after dividing them manually, regarding their onsets and unsilent parts. Therefore, some instrument families, particularly the Strings, are seen to require more samples in this dataset and trusted segmentation strategies would be beneficial to determine convenient audio excerpts from the examined audio files, especially dealing with polyphonic recordings. Also, in the project *So Easy To Love You*, another instance of name-function conflict over instrument usage is observed, which happened because 08_Cello.wav was played with pizzicato technique and reminded a double bass instrument that the ML algorithm approved of.

The Wind instrument family has 44 instruments, only 11 of which were classified correctly. The project *River of The White Gloom* (from the Pop) has two trumpet tracks, and they were previously identified as a member of the Drums&percussion and Keyboard families. After manually dividing these audio files into 10 and 12 parts and then classifying them with the ML algorithm again, this time, the classification accuracy has substantially improved at a rate of 90.0% and 75.0%, respectively. Once more, the importance of the excerpt algorithm that be provided in MixPrep is detected to attain better classification results.

Another critical point is that the classification accuracy of an audio file significantly reduces due to high leakage and low SNR, which frequently appear in live recording situations. The audio tracks from the projects *Corine Corine* and *Song of India* (Jazz) are both observed to suffer from the higher cross-talks in different levels in different

parts of the song. The cross-talks add another level of complexity while the algorithm is solving solves the instrument recognition problem.

5.5 Strategies for Improving the Accuracy

Some important points while approaching the design and implementation of instrument classification methods are indicated for music production steps.

5.5.1 Dataset creation

The first important point is to build a higher quality dataset. The dissertation dataset was constrained using the Apple Jam Pack library consisting of various lengths and numbers of audio loops (in Table 5.1). A higher number of class observations in a dataset can increase the chance to attain more accurate classification results. However, just before the dataset creation, the process of reviewing and eliminating unsuitable content of each instrument family one by one led to a reduction in the number of observations. This situation explains the low PPV results for Keyboard, Wind and especially Strings classes in this dissertation dataset.

Nonetheless, the content sizes of classes in a dataset cannot guarantee better classification rates alone. A suitable and wide variety of audio file organization strategies is necessary to build better datasets and ML models implementing instrument classification methods into an application for real-life scenarios. The organization and coverage of classes (targets) depend on the ML strategy to solve a problem. It is pretty challenging to attain a model with a higher generalization level, especially in dealing with complex problems such as instrument classification.

5.5.2 Genre-specific datasets

In the experiment, the numbers of instrument family members are not equal and change from one genre to another. For example, despite the Jazz genre having 41 and the Pop having only three instruments suited to the Wind family, the other genres do not have any tracks that belong to the wind class. Again, the Pop genre has 33, and Jazz has only three strings; in contrast, the others have none. In some genres, some audio families may not seem essential or have the priority to meet the expectations of the arrangement. So it brings the idea that creating genre-specific datasets will facilitate instrument family estimation.

The available dataset has six instrument families, but MixPrep's machine learner creator is available whenever a new dataset and SVM-based model is wanted to be built by users. A dataset may be genre-specific (Pop, Rock, Jazz, etc.), family-based (like the dataset of this dissertation), or relative-based, which consists of instruments in the same family (e.g., viola, violin, cello, double bass). The datasets can have many classes (like instrument families) and are saved as an excel file for further investigation; however, MixPrep was restricted to using ten subgroups as instrument families to create a Reaper session. In a different scenario, users can create many datasets suitable to their needs for regular mixing workflows. In that case, cumbersome audio editing will be required to remove redundant parts in audio files (like silences) because the software has no segmentation algorithm.

5.5.3 Instrument denotations and functions in the arrangement

Another point appears when audio file denotations contradict their sound contents, which has particularly been observed when reviewing the audio files in the Keyboards family from the Electronic/Dance genre. This contradiction creates a situation that causes false interpretations of the classification results regarding only the denotations in the first place. So, the intended usage of instruments may not be compatible with the definitions or functions of audio files in a multitrack project. The discrepancies between contents and description of some audio files are observed particularly in the synth-tagged tracks of the Electronic/Dance genre.

5.5.4 The excerpt algorithm

The audio tracks should be short enough yet present the recorded instruments well for the loop-based instrument classification approach. Because of that, the MixPrep takes two-second audio excerpts from the provided multitrack files before importing them to the GUI. The two-second excerpt algorithm that is based on the onset detection algorithm demonstrated in Librosa's audio library documentation was considered necessary to eliminate redundant parts (especially silences) of the provided audio tracks to increase classification speed. However, when reviewing the excerpts during the experiments, it has been observed that the algorithm occasionally extracts the audio's unsuitable or less suitable parts, which may cause poor classification results. Manual slicing is observed to have improved the audio file classification success.

Consequently, a better excerption algorithm will increase the classification performance.

5.5.5 Leakage and low SNR levels

Some multitrack projects like *Corine Corine* in the Jazz genre were recorded in live conditions, which caused severe leakage problems between the audio tracks and negatively affected the accuracy of the instrument recognition. Since MixPrep does not use any source separation method or algorithm to extract the salient instrument sound from the provided audio tracks, leakage leads to a change in the temporal and spectral contents of the track. This leads to an increase in the misclassification probability in that situation. Similarly, wrong class estimation probability increases when the audio file is recorded or processed with audio effects or has a low SNR level.

6. CONCLUSION

This dissertation presented an assistant software that can do some basic tasks of multitrack mix preparation. The motivation was to investigate how and to what level of the mix preparation could be automated or at least reach a certain level of automatization. Thus, music production becomes less labor-intensive by the intelligent system taking over the technical complexity of the workflow. Intelligent music production which is the field devoted to solving the technical struggles challenges of music creation with clever tools inspired this study. Knowledge and the literature of ML-based instrument recognition with the implementation of available ACA methods were the core components to design the intelligent feature of the assistant software.

Some preliminary decisions had to be taken to conceptualize the mix-preparation assistant software. The first decision was to determine the music creation platform to integrate the software. Most DAWs are proprietary software that allows no or limited controls over its functionalities. Since the projected software demands extensive control over the DAW functionalities to conduct basic mix preparation requirements which are organizing, grouping and coloring the audio files, open-source DAW alternatives seemed appropriate. The second one was to determine the implementation of the assistant software, whether it is a standalone or plug-in type. Then, the last objective was to find a suitable ML library, then embed this library into the software to execute the (aforementioned) mix preparation processes automatically.

According to the objectives mentioned above, Reaper is the most suitable DAW due to having fully open-source structure and supporting Python scripting language, which is well-known for creating ML solutions. The advantage of the Reaper was that the ReaScript extension allows controlling most of its API functions remotely, which means the mix-preparation software has the opportunity to be developed as a stand-alone application with Python IDE. After determining the development platform, the researcher considered the functionalities and layouts of the GUI of the mix preparation assistant software. The software design philosophy was to build a flexible hybrid software that allowed users to work with automatic (via machine learning) and manual

settings. Hence, users can intervene in subgroup creation and organization processes even though machine learning has already predicted instrument families of each audio file. Also, users can manually create a mixing session by changing subgroup contents, predefined colors, and names without deploying the ML-based subgrouping algorithm. The properties of the GUI were developed regarding this philosophy. As a result, a mix preparation software prototype (MixPrep) emerged.

Every supervised machine learning application requires an appropriate dataset for classification or regression. So, the software has an ability that allows users to create custom datasets with the combinations of different audio features and settings, along with building an ML model that examines the instrument classes of provided audio files. Therefore, MixPrep uses the pyAudioAnaysis ML library for instrument recognition and ML model creation. The model creator can extract 35 audio features in three different statistics and build an SVM-based machine learning model for later usage. In this dissertation experiment, a 5276x106 (included the target instrument family) dimensional dataset was created with the provided audio files. Since the automatic subgrouping idea aims to guess only the instrument families of audio tracks, designing the dissertation dataset focused on the loop-based solution instead of the single-note-duration instrument samples approach. Because the functions of instruments in music production are more important than their specific instrument family most of the time, the intended artistic perspective of the music creation process leads the producer to perceive musical instruments not just as pure instruments that belong to a specific instrument family but also their functions arrangement wise. For this reason, Apple Jam Packs audio library seemed beneficial for building this type of dataset for instrument family estimations regarding its content organization.

Although MixPrep's ML model creator can model estimation while building a dataset, MATLAB was also used to double-check the accuracy and legitimacy of the available dataset by comparing a wide variety of machine learning models. As a consequence, MATLAB approved that the support vector machine algorithm was the ideal classification method for the dissertation dataset by a 95.1% accuracy rate. Even though the pyAudioAnalysis library supports many different ML models, only the SVM algorithm was included in the software for simplicity.

After determining the dataset and machine learning model, 80 multitrack projects selected from the Mixing Secrets website in four genres (Pop, Rock, Jazz, and

Electronic/Dance) were obtained to test the instrument recognition performance of the MixPrep. Consequently, 1428 audio tracks have been examined in total through 80 multitrack projects.

Table 5.15 (Section 5.4) shows average performances of overall, genre-specific and instrument families specific for 80 multitrack projects. The overall performance rate was detected to be 64.71% for all instrument families. Regarding averages of the instrument families, the best recognition rate was 68.52% for the Rock genre. Regarding averages of the genres, the best instrument family estimation average was 84.80% for Drums&percussions. Detailed results are given in the Appendices. Appendix B shows the names of included multitrack projects and their success rates for each genre. Appendix C, D, E, and F show instrument family predictions of each audio file regarding individual multitrack project files from the Pop, Rock, Jazz, and Electronic/Dance genres, respectively.

The results indicate that the dissertation dataset and model can recognize drums and percussive sounds at higher rates, at least 82.50%, without showing extreme results related to the genres. Nonetheless, diverse recognition success rates for instrument families were observed among the different genres. For example, the Bass family, which had the second average successful classification rate among the instrument families, ranked 79.49% in the Rock genre; yet the success rate degraded 51.43% in the Electronic/Dance genre although they had almost the same numbers of bass instrument tracks. Moreover, the same irregularities were seen through the Keyboards family results over the genres, which is interestingly 32.43% for Jazz yet 5.56% for the Rock genre.

Essential points have been mentioned to guide researchers who aim to design intelligent systems and implement them into music production steps similar to the dissertation research. These points were gathered under these subtopics: dataset creation, genre-specific datasets, instrument denotations, functions in the arrangement, audio excerpt algorithm, leakage, and low SNR levels.

In conclusion, a machine learning-based automatic multitrack premixing software prototype has been introduced in this dissertation. This software works coordinatively with the Reaper DAW and creates a multitrack mix session by creating subgroups and changing their properties manually, automatically, or both, which means providing

complete control over the session creation thanks to a hybrid development paradigm. The software utilizes a machine learning-based instrument recognition model to classify multitrack audio files by a dedicated audio dataset. Again, this dissertation demonstrated the loop-based audio dataset method as a novel dataset creation approach for single-source instrument recognition tasks. It has also been observed that the SVM-based machine learning algorithm provides the best classification performance after examining this dataset with many different machine learning algorithms. The software also provides a machine learning creator module that allows users to create their custom audio datasets and SVM-based models for further usage and experimentation.

The mix preparation requires quite a lot of time, considering a musical album generally consists of ten pieces, and each piece has tens of audio tracks. Indeed, as explained in this dissertation, the mix preparation steps (track organization, style dependent, and edit-based) cause an inevitable workload which reduces the engineers' precious energy, which is expected ideally consumed on artistic decisions during the mix. Undoubtedly, automatic mix preparation solutions and their implementations into modern DAWs will provide a comfortable working experience for mixing engineers.

The other mix preparation procedures which are not covered in this study, such as audio alignment, phase correction (for example, for drums track), customization of subgrouping, setting up audio busses over a signal route, and then adding audio plugins on them automatically, creating markers to specific parts of the piece (e.g., intro, verse, chorus), providing a rough mixing, and so on, all have potential to be automatized by the ML and AI with appropriate methodologies.

For future works, MixPrep premixing assistant can be improved and become a tool to create fully customized DAW sessions. Especially for mixing, this type of assistant can be improved to make rough mixes, and genre-specific audio effect suggestions. To make the production steps easier and quicker, the music creation companies can pursue a similar methodology that this dissertation has and bring intelligent solutions for their DAWs.

It is a well-known fact that artists always seek new sonic possibilities in music. We frequently see that the functionality of an instrument in a musical arrangement does not always match the instrument's heritage and history, so the arrangement-wise track functionalities must be evaluated carefully, and the automatic session preparation

methodologies designed in this manner. New approaches for creating audio subgroups by considering the arrangement-wise instrument functionalities are worth investigating.

Deep learning methods have a higher utilization potential for music-related research fields. In this regard, this research can be maintained further with a suitable and enlarged audio library. Obviously, we need more extensive datasets for DL applications, yet the sonic diversity of music-related applications is immense, and it is hard to foresee how their variations occur in real situations. Despite that, we have a few useful tools for alleviating dataset creation difficulties. The first is the audio loops we prefer to use in this dissertation. Many commercial and non-commercial audio loops are available that researchers can use. The second solution is to use MIDI and VST libraries to create vast datasets. However, this may lead to another challenge requiring an extensive workload for researchers to gain massive datasets. A new research opportunity appears to work on automatic solutions that can create musical phrases using VST technologies with available MIDI repertories for attaining new audio datasets.

Moreover, attaining valuable knowledge about sound engineers' habits and behaviors in the production steps and logging them as parameters will provide new opportunities for designing and developing automatic solutions for music production. We can bring more accurate automatic solutions for mixing, mastering, sound design, semantic audio, and similar realms. At that point, Web Audio API and open-source DAWs (like Reaper) will facilitate data acquisition from sound engineers' workflow and retrieve their preferences on mixing and even any production step. With further research and developments, I hope IDAWs (Intelligent Digital Audio Workstations) will emerge soon.

REFERENCES

Alexander, P. J. (1994). New Technology and Market Structure: Evidence from the Music Recording Industry. *Journal of Cultural Economics*, *18*, 113-123.

Alpaydin, E. (2014). *Introduction to Machine Learning*. (3rd Edition) London: MIT Press.

Azarloo, A., & Farokhi, F. (2012). Automatic musical instrument recognition using K-NN and MLP neural networks. Proceedings. *2012 4th International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN,* Phuket Thailand, August 23.

Bhalke, D. G., Rama Rao, C. B., & Bormane, D. S. (2011). Dynamic time warping technique for musical instrument recognition for isolated notes. *2011 International Conference on Emerging Trends in Electrical and Computer Technology, ICETECT,* Nagercoil, India, May 02.

Bhalke, D. G., Rao, C. B. R., & Bormane, D. S. (2014). Stringed musical instrument recognition using fractional fourier transform and linear discriminant analysis. *Proceedings of the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques*, *ICICT*, Ghaziabad, India, 07-08 February.

Bishop, C. M. (2006). *Machine Learning and Pattern Recogniton (In Information Science and Statistics).* New York: Springer-Verlag.

Brown, J. C. (1998). Computer identification of wind instruments using cepstral coefficients. *The Journal of the Acoustical Society of America*, *103(5)*, 1889–1890. https://doi.org/10.1121/1.422375.

Burred, J. J., Röbel, A., & Sikora, T. (2009). Polyphonic musical instrument recognition based on a dynamic model of the spectral envelope. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Taipei, Taiwan, 19-24 April.

C., Y., Shin, & Xu, C. (2009). *Intelligent Systems: Modeling, Optimization, and Control*. CRC Press.

Costantini, G., Rizzi, A., & Casali, D. (2003). Recognition of musical instruments by generalised min-max classifiers, *2003 IEEE XIII Workshop on Neural Networks for Signal Processing*, Toulouse, France, 17-19 September.

De Man, B., Stables, R., & Reiss, J. D. (2019). *Intelligent Music Production*. Focal Press.

Diment, A., Heittola, T., & Virtanen, T. (2013). Semi-supervised learning for musical instrument recognition. *21st European Signal Processing Conference (EUSIPCO 2013),* Marrakech, Morocco, 09-13 September.

Eric, G., Hassan, H., & Bahoura, E. M. (2012). Hierarchical parametrisation and classification for musical instrument recognition. *2012 11th International Conference on Information Science, Signal Processing and Their Applications (ISSPA)*, Montreal, QC, Canada, 02-05 July.

Eronen, A. (2003). Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs. *7th International Symposium on Signal Processing and Its Applications* (*ISSPA)*, Paris, France, July 4.

Eronen, A., & Klapuri, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Istanbul, Turkey, 05-09 June.

Essid, S., Richard, G., & David, B. (2006). Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, *14(4),* 1401–1412.

Ezers, E., & Doroshin, S. Y. (2021). Musical Instruments Recognition App. 2021 *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)* 2021, Petersburg, Moscow, Russia, 26-29 January.

Fanelli, A. M., Caponetti, L., Castellano, G., & Buscicchio, C. A. (2004). Content-based recognition of musical instruments. *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Rome, Italy, 18-21 December.

Garcia, H. F., Aguilar, A., Manilow, E., & Pardo, B. (2021). Leveraging Hierarchical Structures for Few-Shot Musical Instrument Recognition. Retrieved January 10, 2022, from https://arxiv.org/abs/2107.

Giannakopoulos, T. (2015). pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. PLOS ONE, *10(12)*, e0144610. Retrieved May 20, 2011, from https://doi.org/10.1371/journal.pone.0144610.

Giannakopoulos, T., & Pikrakis, A. (2014). *Introduction to Audio Analysis: A MATLAB Approach.*Oxford: Academic Press.

Giannoulis, D., & Klapuri, A. (2013). Musical Instrument Recognition in Polyphonic Audio Using Missing Feature Approach. *IEEE Transactions on Audio, Speech, and Language Processing, 21(9)*, 1805–1817. https://doi.org/10.1109/TASL.2013.2248720.

Gunasekaran, S., & Revathy, K. (2008). Fractal dimension analysis of audio signals for Indian musical instrument recognition. *International Conference on Audio, Language and Image Processing (ICALIP)*, Shanghai, China, 07-09 July.

Gururani, S., Sharma, M., & Lerch, A. (2019). An attention mechanism for musical instrument recognition. *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Deft, Netherlands, November 4-8, 83-90.

Hepworth-Sawyer, R., & Golding, C. (2010). *What is Music Production?: A Producers Guide: The Role, the People, the Process*. Oxford: Focal Press.

Herrera-Boyer, P., Peeters, G., & Dubnov, S. (2003). Automatic classification of musical instrument sounds. Journal of New Music Research, *21(1)*, 3–21, DOI: 10.1076/jnmr.32.1.3.16798.

Kaminskyj, I., & Czaszejko, T. (2005). Automatic recognition of isolated monophonic musical instrument sounds using kNNC. *Journal of Intelligent Information Systems*, *24(2–3)*, 199–221. https://doi.org/10.1007/s10844-005-0323-7.

Kaminskyj, I., & Materka, A. (1995). Automatic source identification of monophonic musical instrument sounds. *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, 27 November.

Kaminskyj, I., & Voumard, P. (1996). Enhanced automatic source identification of monophonic musical instrument sounds. *Australian New Zealand Conference on Intelligent Information Systems*, 18-20 November.

Kitahara, T. (2010). Preface, In Z. W. Ras & A. Wieczorkowska (Eds.) *Advances in Music Information Retrieval* (Vol. 274, pp. 5-9). Berlin, Heildelberg: Springer-Verlag.

Kitahara, T., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. (2005). Instrument identification in polyphonic music: Feature weighting with mixed sounds, pitch-dependent timbre modeling, and use of musical context. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005),* London, United Kingdom, 11-15 September.

Kitahara, T., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. (2006). Instrogram: A new musical instrument recognition technique without using onset detection nor F0 estimation. *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP)*, Toulouse, France**,** 14-19 May.

Kothe, R. S., Bhalke, D. G., & Gutal, P. P. (2016). Musical instrument recognition using k-nearest neighbour and Support Vector Machine. *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology, ICAECCT,* Pune, India, 02-03 December.

Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2019). *Data Science and Machine Learning. In Data Science and Machine Learning*. Boca Raton, FL, USA: CRC press.

Kursa, M. B., & Wieczorkowska, A. A. (2014). Multi-label ferns for efficient recognition of musical instruments in recordings. In: Andreasen, T., Christiansen, H., Cubero, JC., Raś, Z.W. (eds) *Foundations of Intelligent Systems*. ISMIS 2014. Lecture Notes in Computer Science, Vol 8502. Cham: Springer.

Lee, J., & Chun, J. (2002). Musical instrument recognition using hidden Markov model. *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002*, Pacific Grove, CA, USA ,03-06 November.

Lerch, A. (2012). *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. USA: Wiley-IEEE Press.

Leveau, P., Sodoyer, D., & Daudet, L. (2007). Automatic instrument recognition in a polyphonic mixture using sparse representations. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007),* Vienna, Austria, September 23-27.

Li, H. (2021). Piano Education of Children Using Musical Instrument Recognition and Deep Learning Technologies Under the Educational Psychology. *Frontiers in Psychology*, *12*. DOI: 10.3389/fpsyg.2021.705116

Livshin, A., & Rodet, X. (2009). Purging Musical Instrument Sample Databases Using Automatic Musical Instrument Recognition Methods. *IEEE Transactions on Audio, Speech, and Language Processing, 17(5),* 1046–1051. https://doi.org/10.1109/TASL.2009.2018439.

Lostanlen, V., & Cella, C. E. (2016). Deep convolutional networks on the pitch spiral for music instrument recognition. *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016,* New York City, USA, August 7-11.

M., & Senior. (2011). *Mixing Secrets for The Small Studio.* Burlington: Focal Press.

Mahanta, S. K., Rahman Khilji, A. F. U., & Pakray, P. (2021). Deep neural network for musical instrument recognition using MFCCs. *Computacion y Sistemas, 25(2),* 351–360. https://doi.org/10.13053/CyS-25-2-3946.

Maliki, I., & Sofiyanudin. (2018). Musical Instrument Recognition using Mel-Frequency Cepstral Coefficients and Learning Vector Quantization. *IOP Conference Series: Materials Science and Engineering, 407(1)*. https://doi.org/10.1088/1757-899X/407/1/012118.

Martin, K. D., & Kim, Y. E. (1998). 2pMU9. Musical instrument identification: A pattern-recognition approach. *Journal of the Acoustical Society of America, 104*, 1768-1768. DOI:10.1121/1.424083

Moffat, D., & Sandler, M. B. (2019). Approaches in Intelligent Music Production. Arts, *8(4), 125*. https://doi.org/10.3390/arts8040125.

Moorer, J. A. (2000). Audio in the New Millennium. *Journal of the Audio Engineering Society, 48(5),* 490–498.

Morvidone, M., Sturm, B. L., & Daudet, L. (2010). Incorporating scale information with cepstral features: Experiments on musical instrument recognition. *Pattern Recognition Letters, 31(12),* 1489–1497. https://doi.org/10.1016/j.patrec.2009.12.035.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series)*. London: MIT Press.

Oppenheim, A. V, & Schafer, R. W. (2009). *Discrete-Time Signal Processing*. (3rd Edition) London: Pearson.

Owsinski, B. (2014a). *The Mastering Engineer's Handbook*. (3rd Edition) Boston: Cengage Learning PTR.

Owsinski, B. (2014b). *The Mixing Engineer's Handbook.* (3rd Edition) Boston: Cengage Learning PTR.

Proakis, J. G., & Manolakis, D. G. (1996). *Digital Signal Processing*. (3rd Edition) New Jersey: Prentice Hall.

Rajesh, S., & Nalini, N. J. (2020). Musical instrument emotion recognition using deep recurrent neural network. Procedia Computer Science, 167(Iccids 2019), 16–25. https://doi.org/10.1016/j.procs.2020.03.178.

Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication, 17(1–2),* 91–108. https://doi.org/10.1016/0167-6393(95)00009-D.

Roy, P., Roy, S., & De, D. (2020). TMIR: Transient Length Extraction Strategy for ANN-inspired Musical Instrument Recognition. *Proceedings of 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering, (WIECON-ECE), Bhubaneswar, India,* 26-27 December.

Saleh, H. (2020). *The Machine Learning Workshop*. (2nd Edition) Birmingham, UK: Packt Publishing.

Toghiani-Rizi, B., & Windmark, M. (2017). Musical Instrument Recognition Using Their Distinctive Characteristics in Artificial Neural Networks. Retrieved October 28, 2020, From ArXiv: abs/1705.04971.

Weekhout, H. (2019). *Music Production Learn How to Record, Mix and Master Music*. (3rd Edition) New York: Routledge.

Wicaksana, H., Hartono, S., & Wei, F. S. (2006). Recognition of musical instruments. *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2006),* Singapore, 04-07 December.

Wieczorkowska, A. A., Wróblewski, J., Synak, P., & Ślęzak, D. (2003). Application of Temporal Descriptors to Musical Instrument Sound Recognition. *Journal of Intelligent Information Systems*, *21(1)*, 71–93. https://doi.org/10.1023/A:1023505917953.

Wieczorkowska, A., Kubera, E., & Kubik-Komar, A. (2011). Analysis Of Recognition of A Musical Instrument In Sound Mixes Using Support Vector Machines. *Fundamenta Informaticae*, *107(1)*, 85–104. https://doi.org/10.3233/FI-2011-394.

Yu, J., Chen, X. O., & Yang, D. S. (2008). Chinese folk musical instruments recognition in polyphonic music. *International Conference on Audio, Language and Image Processing, (ICALIP 2008),* Shanghai, China, 07-09 July.

Zhang, L., Wang, S., Wang, L., & Zhang, Y. (2013). Musical instrument recognition based on the bionic auditory model. International Conference on Information Science and Cloud Computing Companion (ISCC-C 2013 Guangzhou, China, 07-08 December.

Zlatintsi, A., & Maragos, P. (2013). Multiscale Fractal Analysis of Musical Instrument Signals with Application to Recognition. *IEEE Transactions on Audio, Speech, and Language Processing, 21(4),* 737–748. https://doi.org/10.1109/TASL.2012.2231073.

APPENDICES

APPENDIX A: The best C parameter estimation result and confusion matrix of the trained SVM model.

APPENDIX B: Multitrack projects and success rates for the Pop, Rock, Jazz, and Electronic/Dance genres.

APPENDIX C: Instrument family predictions of audio files for each multitrack project in the genres.

APPENDIX A: The best C parameter estimation result and confusion matrix of the trained SVM model.

Table A.1: Estimating the best C parameter for the dissertation audio dataset. This result appears on the console right after the model was created.

	Bass			Drums&Percs			Guitars			Keyboards			Strings			Wind			Overall	
C	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Acc**	**F1**
0.001	16.7	0.0	0.0	37.4	100.0	54.4	16.7	0.0	0.0	16.7	0.0	0.0	16.7	0.0	0.0	16.7	0.0	0.0	37.4	9.1
0.010	86.5	88.6	87.5	92.7	89.8	91.2	42.1	94.8	58.3	16.7	0.0	0.0	16.7	0.0	0.0	16.7	0.0	0.0	67.7	39.5
0.50	91.2	96.2	93.6	97.6	96.0	96.8	79.3	91.7	85.0	82.2	77.0	79.5	86.3	54.9	67.1	77.8	76.1	77.0	88.6	83.2
1.00	95.2	95.5	95.4	97.0	97.4	97.2	83.7	92.8	88.0	86.5	79.3	82.8	84.5	69.7	76.4	84.5	81.7	83.1	90.9	87.1
5.00	98.8	97.1	98.0	98.0	98.5	98.2	91.0	94.6	92.8	89.3	86.7	88.0	88.0	79.0	83.2	82.0	86.1	84.0	93.8	90.7
10.00	97.4	98.3	97.9	97.9	97.8	97.8	90.5	93.2	91.8	91.1	89.0	90.1	89.8	85.6	87.7	85.9	84.4	85.2	94.1	91.7
20.00	96.3	98.6	97.4	97.6	97.6	97.6	92.0	93.2	92.6	92.3	87.3	89.7	88.5	86.7	87.6	86.3	87.8	87.1	94.2	92.0

Table A.2: A Confusion Matrix for the dissertation audio dataset. These results appear on the console right after the model was created.

True Class \ Predicted Class	Bass	Drums & Percussion	Guitar	Keyboards	Strings	Wind
Bass	15.71	0.15	0.08	0.00	0.00	0.00
Drums & Percussions	0.34	36.47	0.38	0.15	0.00	0.04
Guitar	0.15	0.38	19.62	0.19	0.46	0.27
Keyboards	0.04	0.15	0.57	9.94	0.19	0.49
Strings	0.08	0.15	0.38	0.23	6.41	0.15
Wind	0.00	0.08	0.30	0.27	0.19	6.00

APPENDIX B: Multitrack projects and success rates for the Pop, Rock, Jazz, and Electronic/Dance genres.

Table B.1: The multitrack projects and their success rates for the Pop genre.

Artist/Band	Track Name	Drums & Percussions			Bass			Guitars			Keyboards			Strings			Wind			Total		%
Atlantis Bound	It Was My Fault for Waiting	8	8	100,00%	1	1	100,00%	11	12	91,67%	0	0	-	0	0	-	0	0	-	20	21	**95,24%**
David Tyo	Oh Life	9	10	90,00%	1	1	100,00%	3	4	75,00%	1	1	100,00%	1	1	100,00%	0	0	-	15	17	**88,24%**
David Tyo	Long Way Home	7	7	100,00%	1	1	100,00%	4	5	80,00%	1	2	50,00%	0	0	-	0	0	-	13	15	**86,67%**
Amy Helm & The Handsome Strangers	Rescue Me	10	10	100,00%	1	1	100,00%	1	3	33,33%	0	0	-	0	1	0,00%	0	0	-	12	15	**80,00%**
Benjamin John	Better Way	15	16	93,75%	1	1	100,00%	2	5	40,00%	0	1	0,00%	0	0	-	0	0	-	18	23	**78,26%**
The Wrong'uns	Rothko	4	5	80,00%	1	1	100,00%	3	5	60,00%	0	0	-	0	0	-	0	0	-	8	11	**72,73%**
Ben Carrigan	Well Talk About It All Tonight	11	14	78,57%	1	1	100,00%	1	1	100,00%	1	3	33,33%	6	9	66,67%	0	1	0,00%	20	29	**68,97%**
Finlay	Same Kind of Life	4	7	57,14%	1	1	100,00%	3	3	100,00%	1	3	33,33%	0	0	-	0	0	-	9	14	**64,29%**
BaumXmedia	Koishii	5	5	100,00%	2	2	100,00%	0	0	-	2	9	22,22%	3	3	100,00%	0	0	-	12	19	**63,16%**
David Tyo	Never Ebb But Flow	9	11	81,82%	1	1	100,00%	5	10	50,00%	0	2	0,00%	0	0	-	0	0	-	15	24	**62,50%**
Eduard Semenov	River of The White Gloom	3	3	100,00%	1	1	100,00%	1	2	50,00%	0	0	-	0	0	-	0	2	0,00%	5	8	**62,50%**
Ted Behrman	Convertible	9	9	100,00%	1	2	50,00%	1	3	33,33%	0	4	0,00%	0	0	-	0	0	-	11	18	**61,11%**
David Tyo	Its So Easy to Love You	6	7	85,71%	0	0	-	3	6	50,00%	0	1	0,00%	0	1	0,00%	0	0	-	9	15	**60,00%**
Angelo Boltini	This Town	14	15	93,33%	2	2	100,00%	5	12	41,67%	0	6	0,00%	5	8	62,50%	0	0	-	26	43	**60,47%**
Drumtracks	Ghost Bitch	5	10	50,00%	0	1	0,00%	0	0	-	2	2	100,00%	1	1	100,00%	0	0	-	8	14	**57,14%**
Al James	Schoolboy Fascination	10	15	66,67%	1	1	100,00%	1	4	25,00%	0	2	0,00%	1	1	100,00%	0	0	-	13	23	**56,52%**
Trick Bird	Window	7	9	77,78%	0	2	0,00%	3	6	50,00%	0	0	-	0	0	-	0	0	-	10	17	**58,82%**
Ben Carrigan	Hey Carrie Anne	4	6	66,67%	2	2	100,00%	0	2	0,00%	1	1	100,00%	3	8	37,50%	0	0	-	10	19	**52,63%**
Strobe	Maya	10	11	90,91%	1	2	50,00%	0	4	0,00%	0	5	0,00%	0	0	-	0	0	-	11	22	**50,00%**
James Elder & Mark M Thompson	The English Actor (Full)	5	7	71,43%	0	1	0,00%	1	3	33,33%	2	6	33,33%	0	0	-	0	0	-	8	17	**47,06%**
Numbers of Successful Tracks / Total Tracks		155	185		19	25		48	90		11	48		20	33		0	3		253	384	
Averages				**83,78%**			**76,00%**			**53,33%**			**22,92%**			**60,61%**			**0,00%**			**65,89%**

Table B.2: The multitrack projects and their success rates for the Rock genre.

Artist/Band	Track Name	Drums & Percussions			Bass			Guitars			Keyboards			Strings			Wind			Total		%
Imp Act	The Island of Alsocanla	6	6	100,00%	2	2	100,00%	4	4	100,00%	0	0	-	0	0	-	0	0	-	12	12	100,00%
Blue Lit Moon	Dad's Glad	6	7	85,71%	1	1	100,00%	2	2	100,00%	0	0	-	0	0	-	0	0	-	9	10	90,00%
Candlebox	Happy Pills	10	10	100,00%	2	3	66,67%	1	2	50,00%	0	0	-	0	0	-	0	0	-	13	15	86,67%
Complainiacs	Etc	5	7	71,43%	2	2	100,00%	2	2	100,00%	0	0	-	0	0	-	0	0	-	9	11	81,82%
Decypher	Unseen	9	9	100,00%	6	6	100,00%	7	12	58,33%	0	0	-	0	0	-	0	0	-	22	27	81,48%
Bobby Nobody	Stitch Up	7	7	100,00%	2	2	100,00%	1	4	25,00%	0	0	-	0	0	-	0	0	-	10	13	76,92%
Forkupines	Sleep by The Fire Bloom in Water	19	21	90,48%	1	2	50,00%	3	6	50,00%	0	1	0,00%	0	0	-	0	0	-	23	30	76,67%
The Black Crown	Flames	11	13	84,62%	1	2	50,00%	3	5	60,00%	0	0	-	0	0	-	0	0	-	15	20	75,00%
All Hands Lost	Ambitions	11	12	91,67%	2	2	100,00%	0	4	0,00%	0	0	-	0	0	-	0	0	-	13	18	72,22%
Angels In Amplifiers	I'm Alright	3	5	60,00%	1	1	100,00%	3	3	100,00%	0	1	0,00%	0	0	-	0	0	-	7	10	70,00%
Candlebox	Surrendering	8	10	80,00%	3	3	100,00%	2	6	33,33%	0	0	-	0	0	-	0	0	-	13	19	68,42%
Louis Cressy Band	Good Time	9	14	64,29%	1	1	100,00%	3	3	100,00%	0	1	0,00%	0	0	-	0	0	-	13	19	68,42%
Bill Chudziak	Children Of No-one	10	10	100,00%	0	1	0,00%	3	5	60,00%	1	5	20,00%	0	0	-	0	0	-	14	21	66,67%
The Apprehended	Still Flyin	9	9	100,00%	1	1	100,00%	6	10	60,00%	0	5	0,00%	0	0	-	0	0	-	16	25	64,00%
Ale Lak	Nosso Mundo Deixou De Existir	5	6	83,33%	1	1	100,00%	9	17	52,94%	0	0	-	0	0	-	0	0	-	15	24	62,50%
Cnoc An Tursa	Bannockburn	13	16	81,25%	0	2	0,00%	5	12	41,67%	0	1	0,00%	0	0	-	0	0	-	18	31	58,06%
The Black Crown	Cage	7	11	63,64%	2	3	66,67%	4	9	44,44%	0	1	0,00%	0	0	-	0	0	-	13	24	54,17%
Death of a Romantic	The Well	5	8	62,50%	1	2	50,00%	4	8	50,00%	0	1	0,00%	0	0	-	0	0	-	10	19	52,63%
Jet B	Suit You	6	9	66,67%	1	1	100,00%	0	2	0,00%	0	2	0,00%	0	0	-	0	0	-	7	14	50,00%
DarkRide	Dead Enemies	6	10	60,00%	1	1	100,00%	0	5	0,00%	0	0	-	0	0	-	0	0	-	7	16	43,75%
Numbers of Successful Tracks / Total Tracks		165	200		31	39		62	121		1	18		0	0		0	0		259	378	
Averages				**82,50%**			**79,49%**			**51,24%**			**5,56%**			**-**			**-**			**68,52%**

Table B.3: The multitrack projects and their success rates for the Jazz genre.

Artist/Band	Track Name	Drums & Percussions			Bass			Guitars			Keyboards			Strings			Wind			Total		%
H-owl Project	If I Were A Bell	8	8	100,00%	1	1	100,00%	0	0	-	0	1	0,00%	0	0	-	0	0	-	9	10	90,00%
Lorenzo Price	Changing Things	12	12	100,00%	1	1	100,00%	1	2	50,00%	0	1	0,00%	0	0	-	0	0	-	14	16	87,50%
Patrick Talbot	Upper Hand	13	13	100,00%	1	1	100,00%	3	3	100,00%	0	2	0,00%	0	0	-	0	1	0,00%	17	20	85,00%
Jesper Buhl Trio	What Is This Thing Called Love	4	4	100,00%	0	1	0,00%	0	0	-	1	1	100,00%	0	0	-	0	0	-	5	6	83,33%
Dino On The Loose	Queens Light	10	10	100,00%	1	1	100,00%	0	0	-	3	6	50,00%	0	0	-	0	0	-	14	17	82,35%
Patrick Talbot	Blue	9	11	81,82%	1	1	100,00%	2	2	100,00%	0	1	0,00%	0	0	-	0	0	-	12	15	80,00%
Patrick Talbot	A Reason to Leave	14	17	82,35%	0	1	0,00%	3	4	75,00%	2	2	100,00%	0	0	-	0	0	-	19	24	79,17%
Patrick Talbot	Fool	6	8	75,00%	1	1	100,00%	2	3	66,67%	1	1	100,00%	0	0	-	0	0	-	10	13	76,92%
Maurizio Pagnutti Sextet	Bess	9	9	100,00%	1	2	50,00%	0	0	-	0	2	0,00%	0	0	-	2	3	66,67%	12	16	75,00%
Maurizio Pagnutti Sextet	All The Gin Is Gone (full)	9	9	100,00%	2	2	100,00%	0	0	-	0	2	0,00%	0	0	-	1	3	33,33%	12	16	75,00%
Araujo	The Saga of Harrison Crabfeathers	5	7	71,43%	1	1	100,00%	2	2	100,00%	0	1	0,00%	0	0	-	0	0	-	8	11	72,73%
Alejo Granados	Rumba Chonta	8	9	88,89%	1	2	50,00%	1	1	100,00%	0	0	-	0	0	-	3	6	50,00%	13	18	72,22%
Eddie Garrido	Africa	5	5	100,00%	1	1	100,00%	1	2	50,00%	1	2	50,00%	0	1	0,00%	2	3	66,67%	10	14	71,43%
Eddie Garrido	Scarlett	9	12	75,00%	3	3	100,00%	1	1	100,00%	1	5	20,00%	2	2	100,00%	0	0	-	16	23	69,57%
Patrick Talbot	Set Me Free	8	10	80,00%	0	1	0,00%	1	3	33,33%	1	1	100,00%	0	0	-	0	1	0,00%	10	16	62,50%
Eddie Garrido	5th Floor	3	4	75,00%	1	1	100,00%	0	1	0,00%	1	4	25,00%	0	0	-	0	0	-	5	10	50,00%
Pawel Maciwoda	1 Of The First	9	11	81,82%	1	3	33,33%	1	3	33,33%	0	1	0,00%	0	0	-	2	12	16,67%	13	30	43,33%
Abletones Big Band	Song of India	4	9	44,44%	1	2	50,00%	0	1	0,00%	1	2	50,00%	0	0	-	1	6	16,67%	7	20	35,00%
Abletones Big Band	Corine Corine	5	8	62,50%	0	2	0,00%	0	1	0,00%	0	2	0,00%	0	0	-	0	6	0,00%	5	19	26,32%
Lindy Hip Big Band	The Opener	3	3	100,00%	1	1	100,00%	0	1	0,00%	0	1	0,00%	0	0	-	0	13	0,00%	4	19	21,05%
Numbers of Successful Tracks / Total Tracks		150	176		18	28		18	29		12	37		2	3		11	41		211	314	
Averages				**85,23%**			**64,29%**			**62,07%**			**32,43%**			**66,67%**			**26,83%**			**67,20%**

Table B.4: The multitrack projects and their success rates for the Electronic/Dance genre.

Artist/Band	Track Name	Drums & Percussions			Bass			Guitars			Keyboards			Strings			Wind			Total		%
Galias	Horizon	8	8	100,00%	2	2	100,00%	0	0	-	0	3	0,00%	0	0	-	0	0	-	10	13	76,92%
AM Contra	Heart Peripheral	5	5	100,00%	1	1	100,00%	0	0	-	0	2	0,00%	0	0	-	0	0	-	6	8	75,00%
Albert Kader	Ubiquitous	7	7	100,00%	1	1	100,00%	0	0	-	0	3	0,00%	0	0	-	0	0	-	8	11	72,73%
Mustafa Albazy	Shela Ya Marhaba	17	21	80,95%	1	1	100,00%	3	5	60,00%	4	8	50,00%	0	0	-	0	0	-	25	35	71,43%
Babe Grand	Aiguille Rouge	14	18	77,78%	1	1	100,00%	0	0	-	1	4	25,00%	0	0	-	0	0	-	16	23	69,57%
c:fx	Mathematician	7	8	87,50%	2	2	100,00%	0	0	-	2	6	33,33%	0	0	-	0	0	-	11	16	68,75%
Babe Grand	King Of The Weekend	9	12	75,00%	1	1	100,00%	0	0	-	0	2	0,00%	0	0	-	0	0	-	10	15	66,67%
APZX	Transcention	10	12	83,33%	0	1	0,00%	0	0	-	0	3	0,00%	0	0	-	0	0	-	10	16	62,50%
Albert Kader	Whiptails	8	8	100,00%	0	2	0,00%	0	0	-	0	3	0,00%	0	0	-	0	0	-	8	13	61,54%
Ian Dearden	Terania Creek Walking	6	8	75,00%	0	1	0,00%	0	0	-	1	3	33,33%	0	0	-	0	0	-	7	12	58,33%
J0K3R	SeptemberTrance	6	8	75,00%	2	2	100,00%	0	0	-	2	8	25,00%	0	0	-	0	0	-	10	18	55,56%
Mustafa Albazy	Tgheer	15	15	100,00%	1	2	50,00%	0	0	-	1	15	6,67%	0	0	-	0	0	-	17	32	53,13%
APZX	Lacuna	12	12	100,00%	1	2	50,00%	0	0	-	0	11	0,00%	0	0	-	0	0	-	13	25	52,00%
Skelpolu	Anomalous Weeping (Full)	5	5	100,00%	0	1	0,00%	1	1	100,00%	0	5	0,00%	0	0	-	0	0	-	6	12	50,00%
JP Lantieri	Chardonnay Mystik Vibe Remix	6	6	100,00%	0	1	0,00%	0	0	-	0	6	0,00%	0	0	-	0	0	-	6	13	46,15%
Skelpolu	Entwine	4	4	100,00%	0	2	0,00%	0	0	-	2	7	28,57%	0	0	-	0	0	-	6	13	46,15%
JP Lantieri	Chardonnay	4	6	66,67%	1	2	50,00%	0	0	-	0	4	0,00%	0	0	-	0	0	-	5	12	41,67%
Nervbloc	Slapback	10	11	90,91%	2	7	28,57%	0	0	-	0	9	0,00%	0	0	-	0	0	-	12	27	44,44%
Skelpolu	Cold Strive	5	5	100,00%	1	2	50,00%	0	0	-	0	8	0,00%	0	0	-	0	0	-	6	15	40,00%
Aron Jaeger	Fly High	8	10	80,00%	1	1	100,00%	0	0	-	0	12	0,00%	0	0	-	0	0	-	9	23	39,13%
Numbers of Successful Tracks / Total Tracks		166	189		18	35		4	6		13	122		0	0		0	0		201	352	
Averages				**87,83%**			**51,43%**			**66,67%**			**10,66%**			**-**			**-**			**57,10%**

APPENDIX C: Instrument family predictions of audio files for each multitrack project in the genres.

Table C.1: Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
It Was My Fault for Waiting	01_Kick.wav 02_KickSub.wav 03_SnareUp.wav 04_SnareDown.wav 05_Tom.wav 06_Overheads.wav 07_DrumsRoomMono.wav 08_DrumsRoomStereo.wav	09_BassBI.wav	10_ElecGtr1Mic1.wav 11_ElecGtr1Mic2.wav 13_ElecGtr2Mic1.wav 14_ElecGtr2Mic2.wav 15_ElecGtr2MicDI.wav 16_ElecGtr3Mic1.wav 17_ElecGtr3Mic2.wav 18_ElecGtr3DI.wav 19_ElecGtr4Mic1.wav 20_ElecGtr4Mic2.wav 21_ElecGtr4DI.wav	12_ElecGtr1DI.wav		
Oh Life	01_Kick.wav 02_SnareUp.wav 03_SnareDown.wav 04_Tom1.wav 06_OverheadMono.wav 07_OverheadsStereo.wav 08_DrumsRoom.wav 09_CymbalRolls.wav 10_Tambourine.wav	05_Tom2.wav 12_Bass.wav	11_AcousticGtr.wav 14_ElecGtr1.wav 15_ElecGtr2.wav	13_Piano.wav	16_ElecGtr3.wav 17_Strings.wav	
Long Way Home	01_Kick.wav 02_SnareUp.wav 03_SnareDown.wav 04_Tom1.wav 05_Tom2.wav 06_Overheads.wav 08_Tambourine.wav	09_Bass.wav	10_AcousticGtr1.wav 11_AcousticGtr2.wav 13_ElecGtr2.wav 14_ElecGtr3.wav	12_ElecGtr1.wav 15_Piano.wav	16_Organ.wav	

Table C.1 (continued): Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Rescue Me	Chamber.wav Drums-Hat.wav Drums-Kick in.wav Drums-Kick Out.wav Drums-Overhead.wav Drums-Ride.wav Drums-Snare Bottom.wav Drums-Snare Top.wav Drums-Tom1.wav Drums-Tom2.wav Mandolin-DI.wav Mandolin-Mic.wav Room.wav	Bass-DI.wav	Guitar.wav			
Better Way	02_KickAmb.wav 03_SnareCymbals.wav 04_Tambourine.wav 05_TambourineAmb.wav 06_Shaker.wav 07_FingerClicks1.wav 08_FingerClicks2.wav 09_FingerClicks3.wav 10_FingerClicks4.wav 11_Claps1.wav 12_Claps2.wav 13_Claps3.wav 14_Claps4.wav 15_Claps5.wav 16_Claps6.wav 18_Guitalele.wav (ukulele) 19_AcousticGtr.wav	01_Kick.wav 17_Bass.wav	20_ElecGtr1.wav 22_ElecGtr3.wav 23_Piano.wav av	21_ElecGtr2.wav		

Table C.1 (continued): Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Rothko	01_Kick.wav 02_Snare.wav 04_Overheads.wav 05_DrumsRoom.wav	06_Bass.wav	03_Tom.wav 07_AcGtr1.wav 08_AcGtr2.wav 09_ElecGtr1.wav		10_ElecGtr2.wav 11_ElecGtr3.wav	
Well Talk About It All Tonight	01_Kick.wav 04_SnareSample1.wav 05_SnareSample2.wav 06_HiHat.wav 07_Cymbal.wav 08_CymbalSample1.wav 09_CymbalSample2.wav 10_DrumsRoom.wav 11_Shaker.wav 12_Tambourine.wav 13_Hand Claps.wav 29_Harmonica.wav	02_KickSample.wav 14_Bass.wav	03_Snare.wav 15_AcousticGtr.wav 21_Violin2.wav 27_ViolaSamples.wav 28_CelloSamples1.wav	17_Organ.wav	16_Piano.wav 18_PianoSFX.wav 20_Violin1.wav 22_Viola.wav 23_Cello.wav 24_StringsMix.wav 25_StringsTunedMix.wav 26_ViolinSamples.wav	19_Glockenspiel.wav
Same Kind of Life	01_Kick.wav 02_KickSample.wav 03_SnareSample.wav 07_Cymbal.wav	04_Tom1.wav 05_Tom2.wav 06_Tom3.wav 08_Bass.wav	09_ElecGtr1.wav 10_ElecGtr2.wav 11_ElecGtr3.wav	12_Piano1.wav	14_Organ.wav	13_Piano2.wav
Koishii	01_Kick.wav 02_Snare.wav 03_HiHat.wav 04_Claps.wav 05_ReverseCymbal.wav 12_Piano2.wav 13_Synth1.wav	06_Bass1.wav 07_Bass2.wav		15_Synth3.wav 18_Synth6.wav	08_Strings1.wav 09_Strings2.wav 10_Strings3.wav 16_Synth4.wav	11_Piano1.wav 14_Synth2.wav 17_Synth5.wav 19_Synth7.wav

Table C.1 (continued): Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Never Ebb But Flow	01_KickIn.wav 02_SnareUp.wav 03_SnareDown.wav 06_HiHat.wav 07_Overheads.wav 08_Cymbals.wav 09_CymbalRolls.wav 10_Tambourine.wav 17_ElecGtr2.wav 19_ElecGtr3.wav 22_ElecGtr6.wav 24_SFX.wav	04_Tom1.wav 11_Bass.wav	12_Piano.wav 13_AcousticGtr.wav 15_ElecGtr1.wav 16_ElecGtr1DT.wav 21_ElecGtr5.wav 23_ElecGtr6DT.wav		14_Synth.wav 18_ElecGtr2DT.wav 20_ElecGtr4.wav	05_Tom2.wav
River of The White Gloom	01_CajonMic1.wav 02_CajonMic2.wav 03_Shaker.wav 07_Trumpet.wav	04_Bass.wav 05_AcGtr.wav	06_AcGtrDI.wav	08_MIDITrumpet.wav		
Convertible	01_Kick1.wav 02_Kick2.wav 03_Snare1.wav 04_Snare2.wav 05_HiHat1.wav 06_HiHat2.wav 07_DrumFills.wav 08_Crash1.wav 09_Crash2.wav 11_Bass2.wav 12_Synth1.wav 14_Synth3.wav 15_Synth4.wav	10_Bass1.wav	16_ElecGtr1.wav	17_ElecGtr2.wav	13_Synth2.wav 18_ElecGtr3.wav	

Table C.1 (continued): Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Its So Easy to Love You	02_KickRoom.wav 03_Overhead.wav 04_Cymbals.wav 05_Snaps.wav 06_Tambourine.wav 07_Shaker.wav 14_Banjo2.wav	01_Kick.wav 08_Cello.wav	09_AcousticGtr1.wav 10_AcousticGtr2.wav 12_AcousticGtr4.wav 15_Piano.wav	13_Banjo1.wav	11_AcousticGtr3.wav	
This Town	01_Kick.wav 02_KickSide.wav 03_Snare.wav 04_Tom1.wav 05_Tom2.wav 06_Ride.wav 07_Overheads.wav 08_DrumsPZM.wav 09_DrumsWurstMic.wav 10_DrumsRoom.wav 11_DrumsRoomMono.wav 12_DrumsRoomMonoFX.wav 14_Tambourine2.wav 15_Tambourine3.wav 19_AcGtr1Mic2.wav 22_AcGtr2Mic2.wav 29_ElecGtr6.wav 30_Piano.wav 32_Synth2.wav 33_Synth3.wav 36_SynthSFX2.wav 38_Strings_MicPair2.wav	16_BassAmp.wav 17_BassDI.wav 35_SynthSFX1.wav	20_AcGtr1Mic3.wav 21_AcGtr2Mic1.wav 23_AcGtr2Mic3.wav 25_ElecGtr2.wav 28_ElecGtr5.wav 34_Synth4.wav 39_Strings_MicPair3.wav 43_Strings_SectionMic_Cello.wav		13_Tambourine1.wav 18_AcGtr1Mic1.wav 24_ElecGtr1.wav 26_ElecGtr3.wav 27_ElecGtr4.wav 37_Strings_MicPair1.wav 40_Strings_SectionMic_VlnI.wav 41_Strings_SectionMic_Vln2.wav 42_Strings_SectionMic_Vla.wav 44_Strings_CellosCloseMic.wav	31_Synth1.wav

Table C.1 (continued): Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Ghost Bitch	02_SnareUp.wav 03_SnareDown.wav 04_HiHat.wav 09_Overheads.wav 10_DrumsRoom.wav 11_Bass.wav	01_Kick.wav 06_Tom2.wav 08_Tom4.wav	05_Tom1.wav	12_Piano1.wav 13_Piano2.wav	07_Tom3.wav 14_Strings.wav	
Schoolboy Fascination	02_Snare1.wav 03_Snare2.wav 04_Snare3.wav 05_Claps.wav 06_Hat.wav 07_Cymbal1.wav 08_Cymbal2.wav 09_Ride1.wav 10_Ride2.wav 15_Tambourine.wav 18_AcousticGtr.wav 20_ElecGtr2.wav	01_Kick.wav 11_Tom1.wav 13_Tom3.wav 14_Tom4.wav 16_Bass.wav	12_Tom2.wav 17_Piano.wav 21_ElecGtr3.wav 23_Pad.wav		19_ElecGtr1.wav 22_Strings.wav	
Window	01_Kick.wav 02_HiHat.wav 03_Snare.wav 04_Tambourine.wav 05_Tom.wav 06_Cymbal.wav 07_Clap.wav 14_ElecGtr05.wav		10_ElecGtr01.wav 11_ElecGtr02.wav 13_ElecGtr04.wav 16_SFX1.wav	09_BassDI.wav	08_BassMic.wav 12_ElecGtr03.wav	15_ElecGtr06.wav 17_SFX2.wav

Table C.1 (continued): Instrument family predictions of audio files for each multitrack project in the Pop genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Hey Carrie Anne	01_Snare.wav 02_Shaker.wav 03_Tambo.wav 05_Cymbal.wav 11_Cello.wav 13_ViolinSamples.wav 18_ElecGtr.wav 19_ElecGtr.wav	16_DoubleBassSamples1.wav 17_DoubleBassSamples2.wav	15_CelloSamples.wav	06_Piano.wav 07_Glockenspiel.wav	04_Timp.wav 08_Violin1.wav 10_Viola.wav 12_StringsMix.wav	09_Violin2.wav 14_ViolaSamples.wav
Maya	01_Kick.wav 02_Snare.wav 03_HiHat.wav 04_Tom1.wav 05_Tom2.wav 06_Tambourine.wav 07_Shakers.wav 08_Cymbal.wav 09_SFX1.wav 11_Rain.wav 12_Bass1.wav 16_ElecGtrDI2.wav 19_Synth2.wav 21_Synth4.wav 22_Synth5.wav	13_Bass2.wav	10_SFX2.wav 18_Synth1.wav	14_ElecGtrDI1.wav 17_ElecGtrDI2DT.wav		15_ElecGtrDI1DT.wav 20_Synth3.wav
The English Actor	03_SnareDown.wav 04_HiHat.wav 05_Overheads.wav 06_DrumsRoom.wav 07_Toms.wav 08_Bass.wav 09_SynthPad1.wav 15_ElecGtr1.wav 16_ElecGtr2.wav	01_Kick.wav	02_SnareUp.wav 17_ElecGtr3.wav	10_SynthPad2.wav 13_SynthFX1.wav	11_SynthFuzz.wav 12_SynthLead.wav	14_SynthFX2.wav

Table C.2: Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
The Island of Alsocanla	01_Kick.wav 02_SnareUp.wav 03_SnareDown.wav 04_Tom1.wav 05_Tom2.wav 06_Overheads.wav	07_BassAmp.wav 08_BassDI.wav	09_ElecGtr1.wav 10_ElecGtr1DT.wav 11_ElecGtr2Mic1.wav 12_ElecGtr2Mic2.wav			
Dad's Glad	01_Kick.wav 03_Overheads.wav 04_Tom1.wav 05_Tom2.wav 06_Tom3.wav 07_Tom4.wav	08_Bass.wav	02_Snare.wav 09_AcousticGtr1.wav 10_AcousticGtr2.wav			
Happy Pills	Drums-Floor 2-M81.wav Drums-Floor-M81.wav Drums-Hat-M60.wav Drums-Kick In-M82.wav Drums-Kick Out-647.wav Drums-Over-TF51.wav Drums-Rack Tom-M81.wav Drums-Room-M60.wav Drums-Snare Bot-M81.wav Drums-Snare-M80.wav ELE Guitar2 Dis-M81.wav	Bass-Active DI.wav Bass-Amp-U47.wav	ELE GTR1 Dis-M80.wav		Bass-Amp-M80.wav	

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Etc (full)	01_Kick.wav 02_Snare.wav 03_Overheads.wav 05_Tom1.wav 06_Tom2.wav	08_BassDI.wav 09_BassAmp.wav	04_DrumRoom.wav 10_ElecGtr1.wav 11_ElecGtr2.wav		07_Tom3.wav	
Unseen	01_Kick.wav 02_KickTrigger.wav 03_Snare.wav 04_SnareTrigger.wav 05_Tom1.wav 06_Tom2.wav 07_Tom3.wav 08_Tom4.wav 09_Overheads.wav 27_ElecGtr02DTMic2.wav	10_Bass1DI.wav 11_Bass1Mic1.wav 12_Bass1Mic2.wav 13_Bass2DI.wav 14_Bass2Mic1.wav 15_Bass2Mic2.wav	17_ElecGtr01Mic1.wav 18_ElecGtr01Mic2.wav 20_ElecGtr01DTMic1.wav 21_ElecGtr01DTMic2.wav 23_ElecGtr02DIMic1.wav 24_ElecGtr02DIMic2.wav 26_ElecGtr02DTMic1.wav			16_ElecGtr01DI.wav 19_ElecGtr01DTDI.wav 22_ElecGtr02DI.wav 25_ElecGtr02DTDI.wav
Stitch Up	01_KickIn.wav 02_KickOut.wav 03_SnareUp.wav 04_SnareDown.wav 05_Overheads.wav 06_Room.wav 07_LoTom.wav 10_ElecGtr1.wav 11_ElecGtr2.wav	08_BassDI.wav 09_BassCab.wav 13_ElecGtr4.wav	12_ElecGtr3.wav			

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Sleep by The Fire Bloom in Water	01_KickIn.wav 03_KickTrigger.wav 05_KickSampleRoom.wav 06_SnareUp1.wav 07_SnareUp2.wav 08_SnareDown.wav 09_SnareTrigger.wav 10_SnareSample1.wav 11_SnareSampleRoom1.wav 12_SnareSample2.wav 13_HiHat.wav 14_Ride.wav 15_Tom1.wav 16_Tom2.wav 17_Overheads.wav 18_DrumsSquashMic.wav 19_DrumsRoomClose1.wav 20_DrumsRoomClose2.wav 21_DrumsRoomFar.wav 24_Synth.wav 30_ElecGtr3Mic2.wav	04_KickSample.wav 22_BassDI.wav	02_KickOut.wav 23_BassSansamp.wav 25_ElecGtr1Mic1.wav 26_ElecGtr1Mic2.wav 29_ElecGtr3Mic1.wav	27_ElecGtr2Mc1.wav	28_ElecGtr2Mic2.wav	
Flames	01_KickInMic1.wav 02_KickInMic2.wav 03_KickOut.wav 04_Snare1Up.wav 05_Snare1Down.wav 06_Snare2Up.wav 07_Snare2Room.wav 08_HiHat.wav 11_Overheads.wav 12_DrumsRoomStereo.wav 13_DrumsRoomMono.wav	09_Tom1.wav 10_Tom2.wav 14_BassDI.wav	15_BassAmp.wav 16_ElecGtr1Amp.wav 18_ElecGtr1DTAmp.wav 20_ElecGtr2.wav	17_ElecGtr1DI.wav 19_ElecGtr1DTDI.wav		

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Ambitions(full)	Drum Room.wav Guitar 1.wav Guitar 2.wav Guitar 3.wav Guitar 4.wav Hi-Hat.wav Kick Drum.wav Overhead Left.wav Overhead Right.wav Ride.wav Snare Bottom.wav Snare Top.wav Tom 1.wav Tom 2.wav Tom 4.wav	Bass DI.wav Bass Mic.wav			Tom 3.wav	
I' m Alright (full)	01_Kick.wav 02_Snare.wav 03_Overheads.wav	06_Bass.wav	04_Toms.wav 05_Percussion.wav 07_Piano.wav 08_ElecGtr.wav 09_AcGtr1.wav 10_AcGtr2.wav			
Surrendering	Drums-Floor 2-M81.wav Drums-Floor-M81.wav Drums-Hat-M60.wav Drums-Over-TF51.wav Drums-Rack Tom-M81.wav Drums-Room-M60.wav Drums-Snare Bot-M81.wav Drums-Snare-M80.wav ELE Guitar2 Dis-M81.wav	Bass-Active DI.wav Bass-Amp-M80.wav Bass-Amp-U47.wav	Drums-Kick In-M82.wav ELE GTR1 Dis-M80.wav ELE Guitar2 Clean-TF29.wav		Drums-Kick Out-647.wav ELE Guitar1 Clean-M80.wav ELE Guitar1 Clean-TF47.wav ELE Guitar2 Clean-M80.wav	

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Good Time	01_KickIn.wav 02_KickOut.wav 04_HiHat.wav 05_SnareUp.wav 07_Overheads.wav 11_DrumsAmbience1.wav 12_DrumsAmbience2.wav 13_DrumsAmbience3.wav 14_Rainstick.wav 19_Hammond.wav	03_KickSub.wav 08_Tom1.wav 09_Tom2.wav 10_Tom3.wav 15_Bass.wav	06_SnareDown.wav 16_ElecGtr1.wav 17_ElecGtr2.wav 18_ElecGtr4.wav			
Children Of No-one (full)	01_Kick.wav 02_SnareUp.wav 03_SnareDown.wav 04_HiHat.wav 05_Tom1.wav 06_Tom2.wav 07_Tom3.wav 08_Overheads.wav 09_Room1.wav 10_Room2.wav 13_ElecGtr2.wav 16_ElecGtr5.wav 18_Hammond.wav		12_ElecGtr1.wav 14_ElecGtr3.wav 15_ElecGtr4.wav 19_SFXPad1.wav	21_SFXPad3.wav	17_Wurlitzer.wav	11_Bass.wav 20_SFXPad2.wav
Still Flyin (full)	01_Kick.wav 02_SnareUp.wav 03_SnareDown.wav 04_HiHat.wav 05_Tom1.wav 06_Tom2.wav 07_ChinaCymbal.wav 08_Overheads.wav 09_DrumsRoom.wav 22_Synth2.wav	10_Bass.wav	11_ElecGtr01.wav 12_ElecGtr02.wav 14_ElecGtr04DI.wav 17_ElecGtr07.wav 18_ElecGtr08.wav 19_ElecGtr09.wav 25_Piano.wav	13_ElecGtr03DI.wav 16_ElecGtr06.wav	20_ElecGtr10.wav 21_Synth1.wav 23_Synth3.wav 24_Synth4.wav	15_ElecGtr05DI.wav

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Nosso Mundo Deixou De Existir (full)	01_Kick.wav 02_Snare.wav 03_HiHat.wav 05_Overheads.wav 06_DrumsRoom.wav 13_ElecGtr02DI.wav 15_ElecGtr02DTDI.wav 17_ElecGtr03Amp2.wav 22_ElecGtr04DTDI.wav	04_Toms.wav 07_Bass.wav	08_ElecGtr01.wav 10_ElecGtr01DT.wav 12_ElecGtr02.wav 14_ElecGtr02DT.wav 16_ElecGtr03Amp1.wav 18_ElecGtr03DI.wav 19_ElecGtr04.wav 21_ElecGtr04DT.wav 23_ElecGtr05.wav	09_ElecGtr01DI.wav 11_ElecGtr01DTDI.wav 20_ElecGtr04DI.wav	24_ElecGtr05DI.wav	
Bannockburn	01_Kick1.wav 02_Kick2.wav 03_SnareUp.wav 04_SnareDown.wav 05_SnareSample.wav 06_HiHat.wav 07_Ride.wav 08_Overheads.wav 09_DrumsRoom1.wav 10_DrumsRoom2.wav 11_Tom1.wav 13_Tom2.wav 15_Tom3.wav 17_BassDI.wav 18_BassAmp.wav 20_ElecGtr1Mic1.wav 23_ElecGtr2Mic1.wav 29_ElecGtr4Mic1.wav	12_Tom1Sample.wav 14_Tome2Sample.wav 16_Tom3Sample.wav	21_ElecGtr1Mic2.wav 24_ElecGtr2Mic2.wav 26_ElecGtr3Mic1.wav 27_ElecGtr3Mic2.wav 30_ElecGtr4Mic2.wav			19_ElecGtr1DI.wav 22_ElecGtr2DI.wav 25_ElecGtr3DI.wav 28_ElecGtr4DI.wav 31_SynthChoir.wav

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Cage (full)	01_Kick.wav 02_Snare.wav 06_HiHat.wav 07_Ride.wav 08_Overheads.wav 09_DrumsRoomMono.wav 10_DrumsRoomStereo.wav 14_BassFX.wav 19_ElecGtr2DT.wav 20_ElecGtr3.wav 21_ElecGtr3DT.wav	03_Tom1.wav 04_Tom2.wav 12_BassAmp.wav 13_BassDI.wav	18_ElecGtr2.wav 22_ElecGtr4.wav 23_ElecGtr5.wav 24_ElecGtr5DT.wav		05_Tom3.wav	11_SFX.wav 15_Synth.wav 16_ElecGtr1.wav 17_ElecGtr1DT.wav
The Well (full)	01_Kick.wav 03_SnareUp.wav 04_SnareDown.wav 06_Overheads.wav 08_Tom2Sample.wav 10_BassFX.wav 11_ElecGtr01.wav 19_Synth.wav	02_KickSample.wav 07_Tom1Sample.wav 09_BassDI.wav	05_SnareSample.wav 13_ElecGtr02.wav 15_ElecGtr03.wav 16_ElecGtr03DI.wav 18_ElecGtr04DI.wav	12_ElecGtr01DI.wav 14_ElecGtr02DI.wav	17_ElecGtr04.wav	
Suit You	01_Kick.wav 02_Snare.wav 03_HiHat.wav 04_Overheads.wav 05_DrumsRoom.wav 09_Tom4.wav 11_ElecGtr01.wav 12_ElecGtr02.wav 13_Piano.wav 14_Piano render 002.wav	06_Tom1.wav 10_Bass.wav	07_Tom2.wav		08_Tom3.wav	

Table C.2 (continued): Instrument family predictions of audio files for each multitrack project in the Rock genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Dead Enemies	01_KickMic1.wav 02_KickMic2.wav 03_KickSample.wav 06_SnareDown.wav 09_Overheads.wav 10_DrumsRoom.wav 12_ElecGtr01.wav 13_ElecGtr01DT.wav 14_ElecGtr02.wav 15_ElecGtr03.wav 16_ElecGtr04.wav	11_Bass.wav	04_SnareUpMic1.wav 05_SnareUpMic2.wav 07_SnareSample.wav		08_Tom.wav	

Table C.3: Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
If I Were A Bell	01_KickMic1.wav 02_KickMic2.wav 03_Snare.wav 04_HiHat.wav 05_Tom1.wav 06_Tom2.wav 07_Overheads.wav 08_DrumsRoom.wav	09_Bass.wav				10_Piano.wav
Changing Things (full)	01_KickMic1.wav 02_KickMic2.wav 03_SnareUp.wav 04_SnareDown.wav 05_Tom1.wav 06_Tom2.wav 07_Tom3.wav 08_HiHat.wav 09_Ride1.wav 10_Ride2.wav 11_Overheads.wav 12_DrumsRoom.wav	13_Bass.wav	14_Piano.wav 15_ElecGtr1.wav		16_ElecGtr2.wav	
Upper Hand	01_KickIn.wav 02_KickOut.wav 03_KickSample.wav 04_SnareUp.wav 05_SnareDown.wav 06_Tom1.wav 07_Tom2.wav 08_Tom3.wav 09_Overheads.wav 10_DrumsRoom1_Stereo.wav 11_DrumsRoom2_Mono.wav 12_Shaker.wav 13_Tambourine.wav 18_Rhodes.wav	14_Bass.wav	15_AcousticGtr1.wav 16_AcoustiGtr2.wav 17_ElecGtr.wav	20_Horns.wav		19_Piano.wav

Table C.3 (continued): Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
What Is This Thing Called Love	01_KickInside.wav 02_KickBeater.wav 03_Snare.wav 04_Overheads.wav 05_Bass.wav			06_Piano.wav		
Queens Light	01_KickIn.wav 02_KickOut.wav 03_Snare.wav 04_HiHat.wav 05_Tom1.wav 06_Tom2.wav 07_Overheads.wav 08_Congas.wav 09_Shaker.wav 10_Tambourine.wav 15_Synth4.wav 16_Hammond.wav	11_Bass.wav		12_Synth1.wav 14_Synth3.wav 17_ElecPiano1.wav		13_Synth2.wav
Blue	02_KickOut.wav 04_SnareUp1.wav 05_SnareUp2.wav 06_SnareDown.wav 07_HiHat.wav 08_Overheads.wav 09_DrumsRoom1.wav 10_DrumsRoom2.wav 14_Vibes.wav	11_Bass.wav	12_ElecGtr1.wav 13_ElecGtr2.wav 15_Rhodes.wav			01_KickIn.wav 03_KickSub.wav

Table C.3 (continued): Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
A Reason to Leave	02_KickOut.wav 04_SnareUpMic1.wav 05_SnareUpMic2.wav 06_DnareDown.wav 07_HiHat.wav 08_Cymbals.wav 09_DrumsRoom1.wav 10_DrumsRoom2.wav 11_DrumsRoom3.wav 13_DrumsRoom5.wav 14_Tambourine1.wav 15_Tambourine1Room.wav 16_Tambourine2.wav 24_Vibrophone.wav	01_KickIn.wav 03_KickSub.wav 12_DrumsRoom4.wav	17_BassDI.wav 18_ElecGtr01.wav 19_ElecGtr04.wav 20_ElecGtr06.wav	21_ElecGtr07.wav 22_Rhodes.wav 23_ElecPiano.wav		
Fool	03_SnareUp.wav 04_SnareDown.wav 05_HiHat.wav 06_Overheads.wav 07_DrumsRoom1.wav 08_DrumsRoom2.wav	01_KickIn.wav 09_Bass.wav	12_ElecGtr01DT.wav 13_ElecGtr02.wav	10_Rhodes.wav 11_ElecGtr01.wav		02_KickOut.wav
Bess (full)	01_KickIn.wav 02_KickOut.wav 03_SnareUp.wav 04_SnareDown.wav 05_HiHat.wav 06_Tom1.wav 07_Tom2.wav 08_Tom3.wav 09_Overheads.wav 10_BassMic.wav 13_PianoMics2.wav	11_BassDI.wav	12_PianoMics1.wav 15_Trombone.wav			14_Clarinet.wav 16_Trumpet.wav

Table C.3 (continued): Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
All The Gin Is Gone (full)	01_KickIn.wav 02_KickOut.wav 03_SnareUp.wav 04_SnareDown.wav 05_HiHat.wav 06_Tom1.wav 07_Tom2.wav 08_Tom3.wav 09_Overheads.wav 14_Trumpet.wav	10_BassMic.wav 11_BassDI.wav	12_PianoMics1.wav 13_PianoMics2.wav		16_Saxophone.wav	15_Trombone.wav
The Saga of Harrison Crabfeathers	01_KickIn.wav 02_KickOut.wav 03_Snare.wav 06_Overhead1.wav 07_Overhead2.wav	05_Tom2.wav 08_BassDI.wav	04_Tom1.wav 09_ElecGtrMic1.wav 10_ElecGtrMic2.wav			11_Piano.wav
Rumba Chonta	01_Bombo.wav 02_SnareUp.wav 03_SnareDown.wav 04_HiHat.wav 06_Cunono3Sample.wav 07_Paliteo.wav 08_Guasa.wav 10_BassMic.wav 11_Marimba.wav	09_BassDI.wav	12_ElecGtr.wav 13_Saxophone1.wav 14_Saxophone2.wav	05_Cunono.wav	17_Saxophone5.wav	15_Saxophone3.wav 16_Saxophone4.wav 18_Saxophone6.wav

Table C.3 (continued): Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Africa (full)	01_DrumLoop.wav 02_Djembe.wav 03_Percussion.wav 04_Cowbell.wav 05_Cymbals.wav 09_Guitar.wav 14_Trumpets.wav	06_Bass.wav	10_GuitarMarimba.wav	07_Piano.wav		08_Flute.wav 11_Synth.wav 12_StringsChoir.wav 13_Trombones.wav
Scarlett (full)	01_Drums.wav 02_Percussion1.wav 03_Percussion2.wav 04_SnareFills.wav 05_Cymbals.wav 06_Loop1.wav 07_Loop2.wav 09_Cabasa.wav 22_SFX2.wav	10_Bass1.wav 11_Bass2.wav 12_Bass3.wav	13_ElecGtr.wav 14_Piano1.wav 16_Piano3Pad.wav 18_Synth.wav	15_Piano2.wav	08_Maracas.wav 19_Strings1.wav 20_Strings2.wav 23_SFX3.wav	17_ElecPianoPad.wav 21_SFX1.wav
Set Me Free	01_KickIn.wav 03_SnareUp.wav 05_HiHat.wav 06_Overheads.wav 07_DrumsRoom1.wav 08_DrumsRoom2.wav 09_DrumLoop.wav 10_Shaker.wav 13_ElecGtr02.wav	02_KickOut.wav	04_SnareDown.wav 11_Bass.wav 12_ElecGtr01.wav 15_Horns.wav	16_Keys.wav	14_ElecGtr02DT.wa	

Table C.3 (continued): Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
5th Floor (full)	01_Loops.wav 03_Cymbals.wav 04_Percussion.wav 07_ElecGtr.wav	05_Bass.wav		06_ElecPiano.wav	02_Toms.wav	08_Synth1.wav 09_Synth2.wav 10_Synth3.wav
1 Of The First (full)	02_KickMic2.wav 03_KickMic3.wav 04_SnareUp.wav 05_SnareDown.wav 06_HiHat.wav 07_Overheads.wav 08_DrumsRoom.wav 09_Tom1.wav 10_Tom2.wav 16_Bass2.wav 17_Bass3.wav 24_Sax3Mic1.wav 29_Sax4Mic3.wav 30_Piano.wav	11_Tom3.wav 14_ElecGtr.wav 15_Bass1.wav	01_KickMic1.wav 12_AcousticGtrMic1.wav 18_Sax1Mic1.wav 22_Sax2Mic2.wav 23_Sax2Mic3.wav 25_Sax3Mic2.wav	13_AcousticGtrMic2.wav 21_Sax2Mic1.wav	20_Sax1Mic3.wav 26_Sax3Mic3.wav 27_Sax4Mic1.wav	19_Sax1Mic2.wav 28_Sax4Mic2.wav
Song of India (full)	DRUM OVERHEAD STEREO L.wav DRUM OVERHEAD STEREO R .wav E GUITAR.wav KICK.wav PIANO A.wav ROOM C.wav SAX A .wav TRUMPET B.wav UPRIGHT BASS DI.wav	UPRIGHT BASS.wav	ROOM A.wav ROOM B.wav ROOM D.wav ROOM STEREO L.wav ROOM STEREO R.wav SAX B.wav TROMBONE A.wav	PANO B.wav	TROMBONE B.wav	TRUMPET A.wav

Table C.3 (continued): Instrument family predictions of audio files for each multitrack project in the Jazz genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Corine Corine (full)	E GUITAR.wav KICK.wav ROOM A.wav ROOM B.wav ROOM STEREO L.wav ROOM STEREO R.wav SAX A.wav SAX B.wav TROMBONE B.wav TRUMPET A.wav TRUMPET B.wav UPRIGHT BASS.wav		DRUM OVERHEAD STEREO L.wav PIANO A.wav PIANO B.wav ROOM D.wav	UPRIGHT BASS – DI.wav	DRUM OVERHEAD STEREO R.wav ROOM C.wav TROMBONE A.wav	
The Opener (full)	01_Kick.wav 02_Snare.wav 03_Overhead.wav 05_ElecGtr.wav 08_Trumpet2.wav 10_Trumpet4.wav 13_SaxTenor01.wav 16_Trombone1.wav 18_Trombone3.wav	04_BassDI.wav	06_Piano.wav 11_SaxAlto01.wav 15_SaxBaritone.wav 17_Trombone2.wav 19_TromboneBass.wav		07_Trumpet1.wav 09_Trumpet3.wav 12_SaxAlto02.wav 14_SaxTenor02.wav	

Table C.4: Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Horizon	01_Kicks.wav 02_Snares.wav 03_HiHat.wav 04_Toms.wav 05_Shaker.wav 06_SFX1.wav 07_SFX2.wav 08_SFX3.wav 11_Synth1.wav	09_Basses.wav 10_SubBass.wav			13_Piano.wav	12_Synth2.wav
Heart Peripheral	01_Loop1.wav 02_Loop2.wav 03_Loop3.wav 04_Loop4.wav 05_Loop5.wav 07_Synth1.wav	06_BassSynth.wav				08_Synth2.wav
Ubiquitous	01_Kick.wav 02_HiHat1.wav 03_HiHat2.wav 04_Claps.wav 05_SFX1.wav 06_SFX2.wav 07_Shakers.wav 09_Synth1_Unmodulated.wav 11_Synth2.wav	08_Bass.wav 10_Synth2_PitchModulated.wav				

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Shela Ya Marhaba (full)	02_Loop2.wav 03_Loop3.wav 04_Kick1.wav 05_Kick2.wav 06_Clap1.wav 07_Clap2.wav 08_Snare.wav 09_Cajon.wav 11_ExoDrum2.wav 12_Shaker.wav 13_Tambourine.wav 21_Synth2.wav 22_Synth3.wav 26_Synth7.wav 28_SFX1.wav 29_SFX2.wav 30_SFX3.wav 32_SFX5.wav 33_SFX6.wav 34_SFX7.wav	10_ExoDrum1.wav 14_Bass.wav 31_SFX4.wav	01_Loop1.wav 16_ElecGtr2.wav 17_ElecGtr3.wav 18_ElecGtr4.wav 20_Synth1.wav	23_Synth4.wav 24_Synth5.wav 25_Synth6.wav 27_Synth8.wav 35_SFX8.wav	15_ElecGtr1.wav	19_ElecGtr5.wav
Aiguille Rouge	01_Kick.wav 02_KickSub.wav 04_HiHat.wav 05_Clap1.wav 06_Clap2.wav 09_Cymbal1.wav 10_Cymbal2.wav 11_Shaker1.wav 12_Shaker2.wav 13_Percussion1.wav 14_Percussion2.wav 15_SFX.wav 16_Glitch1.wav 18_TapeHiss.wav	17_Glitch2.wav 19_Bass.wav	03_Snare.wav 20_Synth1.wav 21_Synth2.wav 23_Synth4.wav	08_SteelTongueDrum.wav 22_Synth3.wav	07_Taiko.wav	

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Mathematician	01_Kick.wav 02_Snare.wav 03_Clap1.wav 04_Clap2.wav 05_HiHat2.wav 06_HiHat2.wav 07_HiHat3.wav 10_Synth1Dry.wav 15_Synth3Wet.wav	08_Bass1.wav 09_Bass2.wav	14_Synth3Dry.wav	12_Synth2Dry.wav 13_Synth2Wet.wav	11_Synth1Wet.wav	16_SFX1.wav
King Of The Weekend	03_HiHat1.wav 04_HiHat2.wav 05_Snare.wav 06_Percussion1.wav 07_Percussion2.wav 08_Percussion3.wav 10_Percussion5.wav 11_SFX1.wav 13_SFX3.wav 14_Synth1.wav 15_Synth2.wav	01_KickBass.wav	09_Percussion4.wav		12_SFX2.wav	02_Clap.wav
Transcention	01_Kick.wav 02_Snare.wav 03_HatOpen.wav 04_HatClosed1.wav 05_HatClosed2.wav 06_Clap.wav 07_Percussion1.wav 09_Percussion3.wav 11_Percussion5.wav 12_Bass.wav 13_SynthArp.wav 14_SynthLead.wav 16_SFX.wav		08_Percussion2.wav			10_Percussion4.wav 15_ElectroChoir.wav

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Whiptails	01_Loop1.wav 02_Loop2.wav 03_Kick.wav 04_Snare.wav 05_Claps.wav 06_HiHat1.wav 07_HiHat2.wav 08_SFX.wav 09_Bass1.wav 13_Synth3.wav		10_Bass2.wav 12_Synth2.wav			11_Synth1.wav
Terania Creek Walking	02_Snare.wav 03_SnareSample.wav 04_HiHat.wav 05_Overheads.wav 07_Percussion.wav 08_LoopSFX.wav 10_Synth1.wav 11_Synth2.wav	01_Kick.wav 06_Tom.wav	09_Bass.wav	12_Organ.wav		
SeptemberTrance (full)	01_Loop1.wav 02_Loop2.wav 03_Kick.wav 04_HiHat.wav 05_Claps.wav 06_SFX1.wav 13_Synth3.wav 17_Synth7.wav 18_Synth8.wav	07_SFX2.wav 09_Bass1.wav 10_Bass2.wav	12_Synth2.wav 14_Synth4.wav	11_Synth1.wav 15_Synth5.wav	16_Synth6.wav	08_SFX3.wav

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Tgheer	01_Kick.wav 02_Clap1.wav 03_Clap2.wav 04_ElecDrum1.wav 05_ElecDrum2.wav 06_ElecDrum3.wav 07_Snare1.wav 08_Snare2.wav 09_Snare3.wav 10_Hat.wav 11_Shaker.wav 12_SFX1.wav 13_SFX2.wav 14_SFX3.wav 15_SFX4.wav 16_Bass1.wav 19_Synth02.wav 21_Synth04.wav 22_Synth05.wav 24_Synth07.wav 25_Synth08.wav 26_Synth09.wav 27_Synth10.wav 28_Synth11.wav 29_Synth12.wav 30_SynthSFX1.wav 31_SynthSFX2.wav 32_SynthSFX3.wav	17_Bass2.wav	18_Synth01.wav 20_Synth03.wav	23_Synth06.wav		

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Lacuna	01_Kick.wav 02_Snare.wav 03_Clap.wav 04_Hat1.wav 05_Hat2.wav 06_Hat3.wav 07_Hat4.wav 08_Hat5.wav 09_Tom1.wav 10_Tom2.wav 11_Bass1.wav 13_SFX1.wav 14_SFX2.wav 16_Synth02.wav 19_Synth05.wav 21_Synth07.wav 22_Synth08.wav 23_Synth09.wav 25_SynthLead2.wav	12_Bass2.wav	18_Synth04.wav 24_SynthLead1.wav		15_Synth01.wav 17_Synth03.wav	20_Synth06.wav
Anomalous Weeping (Full)	01_Kick.wav 02_Snare.wav 03_HiHat.wav 04_Cymbal.wav 05_SFXHit.wav 06_Bass.wav		07_Harp.wav 11_Synth3.wav 12_Synth4.wav		08_Piano.wav	09_Synth1.wav 10_Synth2.wav

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Chardonnay Mystik Vibe Remix	01_Kick.wav 02_Bass.wav 03_Percussion1.wav 04_Percussion2.wav 05_Percussion3.wav 06_SnareRolls.wav 07_SFX.wav 08_Synth1.wav 09_Synth2.wav 11_Synth4.wav 13_Synth6.wav		10_Synth3.wav			12_Synth5.wav
Entwine (full)	01_Kick.wav 02_Snare.wav 03_HiHat.wav 04_Cymbal.wav 06_SubBass.wav		05_Bass.wav 10_Synth4.wav	07_Synth1.wav 09_Synth3.wav	08_Synth2.wav 12_Synth6.wav	11_Synth5.wav 13_Synth6FX.wav
Chardonnay	01_Kick.wav 03_Percussion1.wav 04_Percussion2.wav 05_Bass.wav 08_Synth2.wav 09_Synth3.wav 11_SFX1.wav	02_KickFX.wav 06_SubBass.wav	12_SFX2.wav		10_Synth4.wav	07_Synth1.wav

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Slapback (full)	01_Loop1.wav 02_Loop2.wav 03_Loop3.wav 04_Loop4.wav 05_Loop5.wav 06_Loop6.wav 07_RevCymbal.wav 12_Bass3c.wav 16_Synth2.wav 17_Synth3.wav 18_Synth4.wav 21_Synth7.wav 24_SFX1.wav 25_SFX2.wav 27_SFX4.wav	08_Bass1.wav 09_Bass2.wav 15_Synth1.wav	10_Bass3a.wav 11_Bass3b.wav 13_Bass4.wav 14_Bass5.wav 19_Synth5.wav 23_Synth9.wav		20_Synth6.wav 22_Synth8.wav 26_SFX3.wav	
Cold Strive	01_Kick.wav 02_Snare.wav 03_Hi-Hat.wav 04_Clap.wav 05_Cymbal.wav 12_Synth3.wav 14_Synth5.wav 15_Synth6.wav	07_BassSub.wav	06_Bass.wav 09_PianoSFX.wav		08_Piano.wav 10_Synth1.wav 11_Synth2.wav 13_Synth4.wav	

Table C.4 (continued): Instrument family predictions of audio files for each multitrack project in the Electronic/Dance genre.

Project Names	Instrument Families/Classes					
	Drums & Percussions	Bass	Guitars	Keyboards	Strings	Wind
Fly High	01_Kick1.wav 03_Snare2.wav 04_Snare3.wav 05_ReverseSnare.wav 06_HiHat.wav 07_Toms.wav 08_ReverseSFX1.wav 10_SFX.wav 12_Synth01.wav 14_Synth03.wav 15_Synth04.wav 17_Synth06.wav 18_Synth07.wav 19_Synth08.wav 20_Synth09.wav 22_Synth11.wav 23_Synth12.wav	11_Bass.wav	09_ReverseSFX2.wav 13_Synth02.wav 16_Synth05.wav 21_Synth10.wav	02_Snare1.wav		

CURRICULUM VITAE

Name Surname **:** İsmet Emre YÜCEL

EDUCATION **:**

- **B.Sc.** **:** 2010, Sivas Cumhuriyet University, Fine Arts, Music Technology
- **M.Sc.** **:** 2014, Dokuz Eylül University, Fine Arts, Music Technology

PROFESSIONAL EXPERIENCE AND REWARDS:

- Research Assistant at Sakarya University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE DISSERTATION:

- **Yücel İ. E.** & Özdemir, T. (2022). Müzik Prodüksiyonunda Akıllı Sistemler ve Bir Otomatik Ses Miksaj Hazırlığı Uygulaması Modeli. İTÜ 250. Yıl Etkinlikleri Türkiye'de Mühendislik ve Mimarlığın 250 Yılı Uluslararası Sempozyumu
- **Yücel, İ. E.** & Özdemir, T**.** (2021). Using Audio Loops for Instrument Family Recognition in Machine Learning Tasks. Porte Akademik Müzik ve Dans Araştırmaları Dergisi, (21), 7-19. Retrieved from https://dergipark.org.tr/en/pub/porteakademik/issue/66105/1033853

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Yücel, İ. E.** (2013). Kontrol Odalarında Tasarım Yaklaşımları, ATMM (Audio Technologies Music and Media) Proceedings, P. 103, Bilkent University, Ankara. (http://www.atmm-conference.org/atmm-2013/).