KU LEUVEN

ARENBERG DOCTORAL SCHOOL
FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING

Acoustic Event Detection: Feature, Evaluation and Dataset Design

Mina Mounir

Supervisor:
Prof. dr. ir. Toon van Waterschoot
Cosupervisor:
Prof. dr. ing. Peter Karsmakers

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Technology (PhD)

September 2020

Mina MOUNIR

Examination committee:
Prof. dr. ir. Bart Lievens, chair
Prof. dr. ir. Toon van Waterschoot, supervisor
Prof. dr. ing. Peter Karsmakers, co-supervisor
dr. ir. Ann Spriet
Prof. dr ir. Marc Moonen
Prof. dr. ir. Toon Goedemé

Prof. dr. Emmanouil Benetos (Queen Mary University of London)
Prof. dr. ir. Tijl De Bie (Ghent University)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Technology

September 2020

© 2020 KU Leuven – Faculty of Engineering Technology
Uitgegeven in eigen beheer, Mina Mounir, Kasteelpark Arenberg 10, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

> *When you reach the end of what you should know,*
> *you will be at the beginning of what you should sense.*

— Kahlil Gibrán, *Sand and Foam*

Here I am at the end of my PhD adventure trying to introduce an important portion of its output, the Thesis. I confess that writing this section, is a magical moment I kept dreaming of it throughout the whole PhD timeline. This is not only because of the fewer constraints on the structure, the language and the content compared to the rest, but mainly because it is the time to "sense". Now It is one of those less frequent moments in (my) life but very pleasant where I give up the control to a deeper me to feel and speak with less scientific and reasoning filters. So I apologize already if my story seems to you more fiction than science but I write those few words about it, wishing that it may help someone starting, proceeding or concluding a looooong adventure *e.g.*, a PhD and to thank the different persons without whom I won't be who I am today.

The whole story started in Lugano when I wanted to quit the engineering field as I felt an increasing need to do something that has to do with art. This meant dropping out the masters in embedded systems design started one year earlier and searching for a more interesting field, for instance music. At that point in time and thanks to three dear friends everything changed. The first, an old friend *Redi* who listened to me and my thoughts for hours, smiled and said "what do you think about postponing the decision one semester later". I needed a faster answer but I accepted to wait and see. In the same week in Florence, I met *Claire*, a lovely Belgian girl, with whom I enjoyed talking and sharing all our thoughts and future preoccupations. One week later, back to Lugano to keep my promise and resume my masters, I had to follow a signal processing course with a very friendly and helpful instructor *Toon*, who happened to be an engineer, a musician and Belgian. From there, the continuation of the story

can be easily guessed by the reader, which was what I have done the last few years. I would have attributed all that to chance – *C'est le hasard !* – but I am happy to believe in a 4^{th} friend who works in the background to show me his unconditional Love.

Grazie Redi per il tuo ascolto e il tuo prezioso accompagnamento per parecchi momenti di confusione. Merci Claire de partager ta vie avec moi et pour ta confiance qu'à deux on peut y arriver peut être moins vite mais sûrement plus loin. Si j'ai pu surmonter plein de moment down pendant la réalisation de cette aventure c'est grâce à toi et au trois petites merveilles *Lia*, *Miguel* et ... (on y arrivera à trouver un nom avant février). Bedankt Toon voor je hulp bij het starten en beleven van een avontuur dat wetenschap en kunst combineert en voor je constante vertrouwen in mij en respect voor de tijd die ik nodig had om verder te komen in mijn doctoraat. Bedankt voor je verschillende adviezen buiten het onderzoeksdomein (bouw, isolatie, keuken en de van tijd tot tijd PhD-therapie).

To discover and enjoy life but more importantly that I'm loved for who I am and not to what I have done, it is thanks to my family whom I miss loads and counted on for a yearly full battery recharge to survive the PhD and the lovely Belgian winter ... mama Mona (always there at the stressy nights towards the end of deadlines), baba Mounir (I doubted many times if I deserve all his proudness of me. I did it because he never lost his confidence in me.) and my adorable little sisters Marine (who followed me to Switzerland. Why not Belgium as a next step?), Marie-claire (my second home for me and my little family when I go back to Egypt) and Monica (my friend and my best book and movie adviser). I cannot forget to thank my brothers who didn't let the separating distance win Bassem, Boo, Luca, Shero, Abraam, Romio, Aminas, Joe and David.

If I missed the sun from time to time, I kept feeling the warmth inside through the continuous support and love from my family in law – "ma belle famille: mounette, pounet & company" – without whom I wouldn't have been able to manage 2(+1) kids, renew a very old house and conclude a PhD. My Belgian Family was larger than what I expected when I landed in Brussels and I'm so grateful to all of its members. You really contributed in my PhD, starting from my old flatmates Gaël, Martin, Nael and Tomaso, my Focolare family (Toni, Paolo, Vale, Pierre, Samuel and Alexander among much others), my Dutch and music lessons friends.

In this adventure, I had the chance to work in different places and meet great colleagues. This started by a very calm place called Geel where I was initiated to Dutch the hard way. A special thanks to my dear friends and translators Gert and Peter who helped me feeling in family even when not understanding

a word. I want to thank everyone I met in Geel and GroepT campuses for the family spirit. If I reached the end of this PhD, it is by far due to the daily motivation I had to come and meet you my dear STADIUS colleagues. Thank you for the nice lunches, coffee breaks, discussions and ping-pong times. Thanks for all the help and support I got with a real friendship spirit.

I want to thank the members of my examination committee for thoroughly reading the manuscript, their questions, suggestions, and examination.

Thanks to all of you because it is in you that I touch God's love in my life.

Mina Mounir
September 2020

Abstract

It takes more time to think of a silent scene, action or event than finding one that emanates sound. Not only speaking or playing music but almost everything that happens is accompanied with or results in one or more sounds mixed together. This makes acoustic event detection (AED) one of the most researched topics in audio signal processing nowadays and it will probably not see a decline anywhere in the near future. This is due to the thirst for understanding and digitally abstracting more and more events in life via the enormous amount of recorded audio through thousands of applications in our daily routine. But it is also a result of two intrinsic properties of audio: it doesn't need a direct sight to be perceived and is less intrusive to record when compared to image or video.

Many applications such as context-based indexing, health monitoring and smart environments, profit from the techniques developed for AED and results are still far from perfect. For instance, automatic music transcription (AMT) usually needs some corrections by expert musicians, and voice-controlled applications often require you to repeat a voice command or a word to your mobile phone before being understood. This is due to the challenging nature of the AED task. It is more than classifying one event into one of a predefined set of classes as it is pointing out a target event from anything else.

In this thesis we focus on two AED applications which, at first sight, seem to be coming from two different worlds. The first is note onset detection (NOD), an atomic component for many music applications like fingerprinting for search engines and recommender systems, digital effects or simply AMT. From its name, the target events of NOD are the starting of musical notes. For the second application, howling detection (HD), the target is more of an artifact than a desired enjoyable event as howling is that sort of beep that shows up when a closed feedback loop is created between a microphone and a loudspeaker which occurs frequently in public address (PA) systems and hearing aids (HA). A HD algorithm is expected to produce some sort of activation function signaling the resonant frequency as soon as the howling starts in order to automatically filter

it out. Surprisingly, both events share a specific time-frequency pattern, hence the key idea behind the spectral sparsity feature suggested in this work.

After introducing AED in Part I, inspired from the work done for NOD, a general 3-step processing scheme is sketched out for detection of pattern-specific events in audio signals. This is followed by a summary of the state-of-the-art methods used for each of the steps. Part I ends by comparing the different metrics and techniques traditionally used for NOD and HD performance evaluation pointing out how unsuitable they are to handle the imbalanced nature of the datasets used in both problems. Moreover, it suggests a framework for a more fair evaluation and more generalized results using precision-recall (PR) curves and k-fold cross-validation scores.

The two main parts of the thesis reside in Parts II and III, discussing the challenges and suggesting possible solutions for the two applications of interest, NOD and HD, respectively. The contributions for each can be divided into three groups following from the problems' solution steps: feature design, annotated dataset generation and evaluation enhancement.

A feature based on spectral sparsity with two flavors, normalised identification of note onset based on spectral sparsity (NINOS^2) and NINOS^2-Transposed (NINOS^2-T), is suggested for respectively detecting note onsets and howling frequencies. When tested on a dataset of synthetically mixed musical note onsets, NINOS^2 outperformed the state-of-the-art NOD feature, Logarithmic Spectral Flux (LSF), for the sustained-strings instruments, pushing the F_1-score to cross the 50 % border. This group of pitched non-percussive instruments is quite challenging as they have softer onsets, *i.e.*, slowly building-up transients. A novel pre-processing step preceding the application of the NINOS^2 detection function, is found to contribute to the performance increase. The pre-processing consists in retaining a subset of frequencies traditionally neglected but found here to be tightly related to onsets. For HD, NINOS^2-T marked a higher average area under the PR curve (PR-AUC) than all the standalone HD features found in literature, for both music and speech examples. The performance of NINOS^2-T remained the highest when restricting the evaluation to early howling detection.

Existing datasets for both problems are relatively limited in terms of quantity and quality. For NOD, the available datasets are mainly manually annotated by two or more experts, limiting their availability due to the expensive annotation process. Moreover the annotation is subjective and note-context dependent. A similar situation exists for HD where datasets are made of recorded and manually annotated howling or sometimes poorly simulated by sinusoidal superposition. Part II starts by introducing a MATLAB tool "Mix-Notes" which is developed for generating automatically annotated NOD datasets. In Part III, a large HD dataset is created by simulating a closed-loop system, using several acoustic

impulse responses (AIRs) to cover a wide range of howling frequencies, and applying the simulated system to different music and speech input files. On top of using those datasets for testing the suggested NOD and HD features, a different NOD experiment is carried out in which a real NOD dataset is augmented using a semi-synthetic dataset, created using the "Mix-Notes" tool, for training a state-of-the-art data-driven Convolutional Neural Network (CNN) model. This is done to overcome the limited availability of annotated real datasets. When running the experiment on piano excerpts, using two different augmentation strategies, preliminary results show better and more stable performance.

To ensure a fair NOD evaluation, a novel parameter, the *overall time shift in annotations*, is proposed in Part II. While consistently lacking in literature when reporting F_1-scores, this parameter is found crucial for making results comparable for different datasets and algorithms. The best-case F_1-score can vary drastically when including this overall time shift in annotations parameter and it is found beneficial to use this parameter as a tunable hyperparameter when training a deep data-driven model on datasets that are annotated differently. The performance of HD features is traditionally compared for a subset of howling candidates using the receiver operating characteristic (ROC) metric. The use of howling candidates is intended to differentiate between howling and signal components and results in a fairly well balanced dataset, yet it excludes the detection of early howling and ringing. To overcome this limitation, in Part III, we suggest a novel HD approach considering all frequency bins as howling candidates. Since this yields a highly imbalanced dataset, for which ROC evaluation has been proven unsuitable, we propose to use the PR and PR-AUC evaluation metrics instead. Moreover, the PR assessment used a grid of equidistant thresholds in order to evaluate the HD feature robustness to threshold variations.

While searching for answers to the different NOD and HD problems, questions never stopped popping up. Part IV revisits some learned lessons, discusses various open questions and suggests some future steps for further research in the presented topics.

Korte Inhoud

Het duurt langer om een stille scene, actie of gebeurtenis te bedenken dan om er een te vinden die geluid produceert. Niet enkel spreken en muziek spelen, maar nagenoeg alles wat rondom ons gebeurt gaat gepaard met of resulteert in één of meerdere geluiden die worden samengevoegd. Daardoor is akoestische eventdetectie (AED) momenteel een van de meest onderzochte topics in audiosignaalverwerking en dit zal wellicht nog enige tijd zo blijven. Dit is het gevolg van de drang om meer en meer gebeurtenissen in onze leefwereld te begrijpen en digitaal te abstraheren via de enorme hoeveelheid opgenomen audio in duizenden applicaties in onze dagelijkse routine. Maar het is ook een resultaat van twee intrinsieke eigenschappen van geluid: het heeft geen directe zichtlijn nodig om waargenomen te worden en het is minder ingrijpend om op te nemen in vergelijking met beeld of video.

Vele toepassingen, waaronder contextgebaseerde indexing, gezondheidsmonitoring en slimme omgevingen, maken dankbaar gebruik van technieken ontwikkeld voor AED, al zijn de resultaten nog verre van perfect. Bijvoorbeeld, automatische muziektranscriptie (AMT) vraagt typisch enkele correcties door muzikale experts en stemgestuurde toepassingen vereisen vaak een herhaling van een stemcommando of trefwoord aan je mobiele telefoon voor die je begrijpt. Dit komt door de uitdagende aard van het AED-probleem. Het gaat verder dan een gebeurtenis classificeren in een vooraf gedefinieerde verzameling van categoriën, aangezien een beoogde gebeurtenis moet kunnen worden onderscheiden van alle mogelijke andere gebeurtenissen.

In dit proefschrift behandelen we twee AED-toepassingen die, op het eerste gezicht, uit twee verschillende werelden lijken te komen. De eerste is nootaanzetdetectie (NOD), een bouwblok van vele muziektoepassingen zoals fingerprinting voor zoekmachines en recommendersystemen, digitale effecten, of eenvoudigweg AMT. Zoals de naam NOD aangeeft, zijn de beoogde gebeurtenissen in dit geval de aanzetten van muzieknoten. In de tweede toepassing, fluittoondetectie (HD), is de beoogde gebeurtenis eerder een

artefact dan een gewenste, aangename gebeurtenis, aangezien we fluittonen willen detecteren die zich voordoen wanneer een gesloten lus ontstaat door terugkoppeling tussen een microfoon en een luidspreker zoals in *public address* (PA) systemen en hoorapparaten (HA). Een HD-algoritme wordt geacht een soort activatiefunctie te produceren die de resonante frequentie aangeeft van zodra fluittonen opkomen, om die vervolgens automatisch weg te filteren. Opmerkelijk genoeg delen beide toepassingen een specifiek tijdsfrequentiepatroon, wat tot het basisidee geleid heeft achter het kenmerk van spectrale spaarsheid dat in dit werd wordt voorgesteld.

Nadat we in Deel I AED introduceren, stellen we een algemeen 3-stapsschema voor om patroonspecifieke gebeurtenissen in audiosignalen te detecteren, geïnspireerd door het bestaande werk rond NOD. Dit wordt gevolgd door een samenvatting van de state-of-the-artmethodes die in elk van de 3 stappen gebruikt wordt. Deel I wordt afgerond met een vergelijking van verschillende metrieken en technieken die traditioneel gebruikt worden voor het evalueren van de NOD- en HD-performantie, waarbij we aantonen in welke beperkte mate deze geschikt zijn voor de ongebalanceerde datasets die we in beide problemen tegenkomen. Daarnaast suggereren we een kader voor een meer faire evaluatie en meer veralgemeenbare resultaten door gebruik te maken van *precision recall* (PR) curves en scores op basis van *k-fold cross validation*.

De kern van het proefschrift vinden we in Delen II en III, waar we de uitdagingen en mogelijke oplossingen bespreken voor de twee toepassingen in ons interesseveld, respectievelijk NOD en HD. De bijdragen voor elk van beide toepassingen kunnen worden onderverdeeld in drie groepen die overeenkomen met de stappen in het algemene oplossingsschema: ontwerp van kenmerken, creatie van geannoteerde datasets en verbetering van de evaluatieprocedure.

We stellen twee varianten voor van een kenmerk op basis van spectrale spaarsheid, *normalised identification of note onset based on spectral sparsity* (NINOS^2) en *NINOS^2-Transposed* (NINOS^2-T), om respectievelijk nootaanzetten en fluittonen te detecteren. Bij testen op een dataset van synthetisch gemixte muzieknootaanzetten, presteert NINOS^2 beter dan het state-of-the-art NOD-kenmerk, *Logarithmic Spectral Flux* (LSF), voor aangehouden strijkinstrumenten met een F_1-score die de grens van 50 % overschrijdt. Die groep van niet-percussieve instrumenten vormt een uitdaging omdat ze zwakkere aanzetten kunnen vertonen, *i.e.*, traag opbouwende transiënten. Een nieuwe voorverwerkingsstap voor het toepassen van de NINOS^2 detectiefunctie blijkt deels aan de basis van deze performantieverbetering te liggen. De voorverwerking bestaat in het weerhouden van een deelverzameling van frequenties die traditioneel verwaarloosd worden maar waarvan we aantonen dat ze sterk verbonden zijn met nootaanzetten. Voor HD geeft NINOS^2-T een hogere gemiddelde oppervlakte onder de PR-curve (PR-AUC) dan

alle *standalone* HD-kenmerken uit de literatuur, zowel voor muziek- als voor spraakvoorbeelden. De performantie van NINOS^2-T blijft de beste wanneer we de evaluatie beperken tot het detecteren van vroege fluittonen.

Bestaande datasets voor beide problemen zijn relatief beperkt in termen van kwantiteit en kwaliteit. Voor NOD werden de bestaande datasets voornamelijk manueel geannoteerd door twee of meer experten, waardoor ze omwille van het dure annotatieproces beperkt beschikbaar zijn. Bovendien is de annotatie subjectief en afhankelijk van de nootcontext. Een vergelijkbare situatie doet zich voor bij HD, waarvoor datasets ofwel bestaan uit opgenomen en manueel geannoteerde fluittonen ofwel uit soms gebrekkige simulaties op basis van sinusoïdale superpositie. In Deel II wordt een MATLAB-tool "Mix-Notes" voorgesteld, die werd ontwikkeld om automatisch geannoteerde NOD-datasets te genereren. In Deel III wordt een grote HD-dataset gecreëerd door een geslotenlussysteem te simuleren met verschillende akoestische impulsresponsen (AIR) die resulteren in een breed bereik van fluittoonfrequenties. Dit gesimuleerde geslotenlussysteem wordt vervolgens toegepast op verschillende muziek- en spraakinputsignalen. Bovenop het gebruik van deze datasets om de voorgestelde NOD- en HD-kenmerken te testen, voeren we ook een ander NOD-experiment uit waarin een echte NOD-dataset wordt uitgebreid met een semi-synthetische dataset, gecreëerd met de "Mix-Notes" tool, met de bedoeling om een state-of-the-art datagedreven *convolutional neural network* (CNN) model te trainen. De bedoeling hiervan is om de beperkte beschikbaarheid van geannoteerde echte datasets op te vangen. Wanneer het experiment wordt uitgevoerd op pianofragmenten, gebruik maken van twee verschillende uitbreidingsstrategiën, tonen preliminaire resultaten een betere en stabielere performantie.

Teneinde een faire NOD-evaluatie te verzekeren, stellen we in Deel II een nieuwe parameter voor, de *overall time shift in annotations*. Terwijl deze parameter consistent ontbreekt in literatuur waarin F_1-scores worden gerapporteerd, blijkt hij cruciaal te zijn om resultaten overheen verschillende datasets en algoritmes te kunnen vergelijken. De best mogelijke F_1-score kan drastisch variëren wanneer deze *overall time shift in annotations* parameter wordt meegenomen, en het blijkt ook voordelig te zijn om deze parameter te gebruiken als een aanpasbare hyperparameter bij het trainen van diepe datagedreven modellen op datasets die verschillend geannoteerd werden. De performantie van HD-kenmerken wordt traditioneel vergeleken voor een deelverzameling van fluittoonkandidaten, gebruik makend van de *receiver operating characteristic* (ROC) metriek. Het gebruik van fluittoonkandidaten is bedoeld om een onderscheid te maken tussen fluittoon- en signaalcomponenten en resulteert in een vrij gebalanceerde dataset, maar laat niet toe om de detectie van vroege fluittonen en *ringing* te evalueren. Om deze beperking te elimineren, stellen we in Deel III een nieuwe HD-aanpak

voor waarin alle frequentiebins als fluittoonkandidaten worden beschouwd. Aangezien dit leidt tot een sterk ongebalanceerde dataset, waarvoor werd aangetoond dat ROC-evaluatie ongeschikt is, stellen we het gebruik van de PR- en PR-AUC-metrieken voor. Bovendien maakt de PR-assessment gebruik van een rooster van uniform verspreide grenswaarden om zo de robuustheid van de HD-kenmerken ten opzichte van variaties in de grenswaarden te evalueren.

Bij het zoeken naar antwoorden voor de verschillende NOD- en HD-problemen ontstonden er telkens ook nieuwe vragen. In Deel IV kijken we terug op de geleerde lessen, bespreken we verschillende open vragen en suggereren we enkele toekomstige stappen voor verder onderzoek in de voorgestelde topics.

Glossary

Acronyms

AA	audio analysis
Acc	accuracy
Adam	adaptive moment estimation
ADC	analog-to-digital converter
AED	acoustic event detection
AEQ	automatic equalization
AFC	acoustic feedback control
AGC	automatic gain control
AIR	acoustic impulse response
AISS	All Instruments Semi-Synthetic
AMT	automatic music transcription
ASP	audio signal processing
BC	binary classification
BLSTM	Bidirectional Long Short-Term Memory
bpm	beats per minute

CASSE	Computational Analysis of Sound Scenes and Events
CF	Complex Flux
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
csv	comma-separated-values
DCT	Discrete Cosine Transform
DF	detection function
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DSP	digital signal processing
DSP	digital signal processor
F_1	F_1-score
FFT	Fast Fourier Transform
FN	false negative
FP	false positive
FPR	false positives rate
fps	frames-per-second
GMM	Gaussian Mixture Model
GTKO	Generalised Teager-Kaiser Operator
HA	hearing aids
HD	howling detection
HDF	howling detection function
HFC	high-frequency content
HMM	Hidden-Markov model
IMSD	Interframe Magnitude Slope Deviation
INOS^2	identifying note onsets based on spectral sparsity

IPMP	Interframe Peak Magnitude Persistence
KNN	K-Nearest Neighbours
LSF	Logarithmic Spectral Flux
MFCC	Mel-Frequency Cepstrum Coefficients
MIR	music information retrieval
MIREX	Music Information Retrieval Evaluation eXchange
MSG	maximum stable gain
MUMS	McGill University Master Samples
NHS	notch-filter-based howling suppression
NINOS^2	normalised identification of note onset based on spectral sparsity
NINOS^2-T	NINOS^2-Transposed
NMF	Non-Negative Matrix Factorisation
NN	Neural Network
NOD	note onset detection
NWPD	Normalized Weighted Phase Deviation
ODF	onset detection function
opm	onsets per minute
P	precision
PA	public address
PAPR	Peak-to-Average Power Ratio
PHPR	Peak-to-Harmonic Power Ratio
PNPR	Peak-to-Neighboring Power Ratio
PP	peak picking
PR	precision-recall
PR-AUC	area under the PR curve

PSS	Piano Semi-Synthetic
PTPR	Peak-to-Threshold Power Ratio
R	recall
RCF	Rectified Complex Flux
RGA	randomly guessing algorithm
RMS	root mean square
RNN	Recurrent Neural Network
ROC	receiver operating characteristic
ROC-AUC	area under the ROC curve
SF	Super Flux
SFM	spectral flatness measure
SMS	Spectral Modeling Synthesis
SNR	signal-to-noise ratio
SpecF	Spectral Flux
STFT	short-time Fourier transform
SVM	Support Vector Machine
TMS	Transient Modeling Synthesis
TN	true negative
TP	true positive
TPR	true positives rate
UBM	Universal Background model
VAD	Voice Activity Detection

Contents

Preface i

Abstract v

Korte Inhoud ix

Glossary xiii

Contents xvii

List of Figures xxi

List of Tables xxv

Part I Introduction 1

1 Introduction 2
- 1.1 What is Audio? . . . 2
- 1.2 Why Audio? . . . 3
- 1.3 Audio analysis and applications . . . 3
  - 1.3.1 Note onset detection . . . 4
  - 1.3.2 Howling detection . . . 7
- 1.4 Contributions . . . 8
- 1.5 Outline of the Thesis . . . 9

2 Acoustic event detection 11
- 2.1 Acoustic Events . . . 12
- 2.2 Brief history of AED research . . . 13
- 2.3 Note onsets and Howling: Spectrally connected . . . 14
- 2.4 General AED scheme . . . 15

2.4.1 Pre-processing . . . 17
2.4.2 Detection Function (DF) . . . 18
2.4.3 Peak Picking (PP) . . . 24
2.5 Performance Evaluation . . . 26
2.5.1 Traditional binary classification evaluation metrics . . . 26
2.5.2 Graph-based evaluation . . . 28
2.5.3 Precision-Recall (PR) curve: a better graph-based evaluation to handle skewed datasets . . . 32
2.5.4 Cross-validation scoring for small datasets . . . 33
2.6 Conclusion . . . 34

Part II Onset Detection **35**

3 Automatic generation of synthetic music and annotations **36**
3.1 Mix-Notes: a tool for NOD dataset generation . . . 37
3.2 Details of generated semi-synthetic datasets . . . 45
3.3 Conclusion . . . 45

4 Note onset detection based on a spectral sparsity measure **49**
4.1 Introduction . . . 51
4.2 Related work . . . 54
4.3 Proposed note onset detection method . . . 56
4.3.1 Idea: transient vs. steady-state components . . . 56
4.3.2 Pre-processing: Coefficients subset . . . 57
4.3.3 ODF: spectral sparsity feature . . . 60
4.3.4 Peak-Picking . . . 64
4.3.5 Onset Detection Parameters . . . 65
4.4 Experiment setup and evaluation . . . 66
4.4.1 Dataset . . . 66
4.4.2 Performance measures . . . 69
4.4.3 Computational complexity and memory usage . . . 70
4.4.4 Parameter tuning . . . 71
4.4.5 Results and discussion . . . 72
4.5 Conclusion and future work . . . 77

5 CNN-based onset detection using synthetic data augmentation **79**
5.1 Introduction . . . 81
5.2 Data-driven modeling and annotation . . . 83
5.2.1 Data-driven NOD with CNN model . . . 83
5.2.2 Manually vs. automatically annotated datasets . . . 84
5.3 Proposed model training strategies . . . 84
5.4 Datasets . . . 86

5.5 CNN Model . . . 89
5.6 Results . . . 90
5.7 Conclusion and future work . . . 92

6 Evaluation of note onset detection algorithms using annotations time shift 95
6.1 Introduction . . . 97
6.2 Onset Detection and Evaluation . . . 100
6.2.1 Onset detection function . . . 100
6.2.2 Peak-Picking . . . 100
6.2.3 Evaluation method . . . 102
6.3 Experimental evaluation . . . 103
6.3.1 Datasets . . . 103
6.3.2 Experiment setup and parameters . . . 106
6.4 Results . . . 106
6.5 Conclusion & Future work . . . 108
6.6 Additional results . . . 109

Part III Howling Detection 111

7 A large and diversified dataset for howling detection 112
7.1 Introduction . . . 112
7.2 Dataset generation . . . 113
7.3 Problematic samples and solution . . . 115
7.4 Conclusion . . . 117

8 Howling detection based on a spectral sparsity measure 119
8.1 Introduction . . . 121
8.2 Problem statement and existing solutions . . . 124
8.2.1 Acoustic feedback problem . . . 124
8.2.2 Notch-filter-based howling suppression (NHS) . . . 126
8.2.3 State-of-the-art howling detection (HD) features . . . 128
8.3 Proposed howling detection feature . . . 131
8.4 Howling detection dataset . . . 134
8.5 Performance evaluation procedure . . . 134
8.6 Simulation results . . . 135
8.7 Conclusions & Future work . . . 144

Part IV Conclusion 147

9 Conclusion and future work 148

9.1 Summary of contributions 148
9.2 Lessons learned 151
9.2.1 Note Onset Detection 151
9.2.2 Howling Detection 153
9.3 Future work 155
9.3.1 Note Onset Detection 155
9.3.2 Howling Detection 157

Bibliography **159**

Curriculum Vitae **169**

Publication List **171**

List of Figures

1.1 (A)ttack, (D)ecay, (S)ustain and (R)elease model (an isolated accordion note). 5

2.1 A music excerpt's spectrogram identifying similarity between note onsets (vertical ellipses) and howling examples (horizontal ellipse). 15
2.2 Solution scheme for AED. 16
2.3 Moving from ROC to PR graphs. 31

3.1 Screenshot of Mix-Notes GUI: a tool for mixing and automatically annotating music (onsets and offsets). 39
3.2 Notes panel. 40
3.3 Onsets panel. 41
3.4 Notes Lib. panel. 41
3.5 Mix Notes panel 43
3.6 Monophonic mixing of a violin excerpt with 3 notes. 43
3.7 Polyphonic mixing of a violin excerpt with 3 notes. 43
3.8 Batch mix panel. 44

4.1 Solution scheme for NOD. An example of the output of the different steps is shown when NOD is applied on an input music excerpt. 54
4.2 Temporal variation of signal energy per frequency bin and of spectral sparsity for three excerpts from different instruments. Note onsets are indicated by vertical black lines. (Top & Middle) *Low-* and *High-energy* STFT log-magnitude coefficients variation. (Bottom) *Low-* and *High-energy* STFT log-magnitude coefficient vector ℓ_1-norm variation. 58
4.2 59

4.3 2-D unit-ball illustration of the relation between the ℓ_2-norm and ℓ_4-norm. 62
4.4 Comparison of onset detection functions (ODFs) and peak-picking results for three example music excerpts. 73
4.4 74

5.1 Comparison of training and validation F_1-score (F_1)-scores for CNN model trained on the All Instruments Semi-Synthetic (AISS) dataset. 87
5.2 Comparison of validation and testing F_1-scores for CNN model trained on synthetic datasets (piano and all instruments). . . . 88
5.3 F_1 scores for the different model training strategies tested on the MAPS_CL real dataset. Every boxplot is summarizing the 5-fold cross-validation test results. 91
5.4 F_1 scores for the differentmodel training strategies tested on the MAPS_AM real dataset. Every boxplot is summarizing the 5-fold cross-validation test results. 92

6.1 General scheme for note onset detection. 98
6.2 Comparison of LogSpecFlux, NINOS^2 and CNN ODFs for a *Cello* excerpt. 101
6.3 F1-score as a function of threshold τ (vertical axis) and shift δ (horizontal axis), averaged per test set. *Circles* and *crosses* respectively represent the best F1-score average with and without considering the overall shift in annotations δ. 105

7.1 Electroacoustic closed-loop (feedback) system. 113
7.2 Gain profile of the forward path transfer function. 114
7.3 Example of unsuitable and suitable signals for the dataset. . . . 116

8.1 Closed-loop (feedback) system model. 125
8.2 Nyquist stability criterion example. 126
8.3 Notch-filter-based howling suppression (NHS) solution scheme. 127
8.4 HD solution scheme: in the proposed approach, the candidate howling component selection is omitted. 128
8.5 Cross-validation PR-AUC results for full and early HD evaluation of the different features when applied on the speech (top) and music (bottom) datasets. 138
8.6 ROC and PR curves for the speech dataset. 142
8.7 ROC and PR curves for the music dataset. 143
8.8 Comparison of PR-AUC statistics for the different test examples per dataset (speech / music) and HD evaluation scenario (full / early). 145

8.9 Comparison of the howling detection functions (HDFs) for (left) a speech example exhibiting a ringing artifact around 800 Hz and (right) a music excerpt exhibiting a low power howling around 260 Hz. 146

List of Tables

2.1 Confusion matrix for event detection problems. 27

3.1 Instruments Grouping 46

4.1 Instruments Grouping 68
4.2 NOD performance measures for *M* dataset 75
4.3 NOD performance measures for *P* dataset 75
4.4 NOD performance measures for *MR8* dataset 76
4.5 NOD performance measures for *PR8* dataset 76

5.1 Dataset properties 88
5.2 CNN structure [34] 90

6.1 Datasets summary 104
6.2 Comparison of best-case F1-score with and without *Overall Shift in Annotations δ Parameter* 107
6.3 Best-case F1-score with and without *Overall Shift in Annotations δ Parameter* for identifying note onsets based on spectral sparsity $(\text{INOS}^2)(\ell_1)$. 110
6.4 Comparison of best-case F1-score with and without *Overall Shift in Annotations δ Parameter* when evaluated on the *strings* instruments of the *PR8* dataset of Chapter 4. 110

7.1 A howling annotation file example. 115

8.1 Optimal parameter values per feature (speech dataset). 139
8.2 Optimal parameter values per feature (music dataset). 140
8.3 Speech dataset evaluation using the best parameterisation per feature. It compares the best F_1, PR-AUC and area under the ROC curve (ROC-AUC) for the full and early evaluations. 141

8.4 Music dataset evaluation using the best parameterisation per feature. It compares the best F_1, PR-AUC and ROC-AUC for the full and early evaluations. 141

Part I

Introduction

Chapter 1

Introduction

"Piano notes falling one after another and forming a beautiful melody, lying on a marvelous background of birds singing in response to a breeze that is finding its way through the tree leaves". That was happening while trying to find the words for introducing this thesis work and that is an infinitely small portion of what an *Audio* signal can hold.

1.1 What is Audio?

The essence of audio are those longitudinal sound waves traveling through air, each holding a different component to form together a whole audio scene. Alternatively, audio is defined as a *signal* - a function of one or more independent variables like space, time, frequency. This latter definition is more convenient when speaking about *processing* which can be generally described as the art of analyzing and/or altering a signal using some specific operations like amplification, filtering, modulation and signal separation. Originally, audio processing was performed almost exclusively in the analog domain. More and more, several audio processing tasks moved to the digital domain both in terms of the signal representation and processing domains [1]. Moving from analog to digital, the audio signal undergoes two discretization steps, one discretizing the independent variable - *i.e.*, time - and the other discretizing the signal's amplitude. The first is called *sampling* and the second is *quantization*. Those two operations are essentially translating the analog voltage mapping the sound wave's compressions and rarefactions measured by a pressure sensor - *i.e.*, a

microphone - to the digital language. They are accomplished by an analog-to-digital converter (ADC) device.

1.2 Why Audio?

Audio signal processing (ASP) is a fast growing research area that continues to investigate new audio related problems. This comes as a consequence of the growing number of digital audio devices and applications used in our daily life. But also, it is a clear result of the different properties of a sound wave especially when compared to its friends, image and video. A first and probably the most powerful property of audio is its ability to traverse obstacles. This added to the cheaper prices of microphone sensors compared to cameras, puts lower restriction on the number of used sensors and their positioning with respect to the source signal. Last but not least, one could intuitively consider microphones less intrusive compared to cameras.

On the other hand, thinking about the established ASP research topics, there is still a big room for performance improvement. This becomes clear when questioning the way we use different audio-based daily applications. For instance, can you think on how many times you needed to repeat a voice command to your smartphone, before having the requested task done for you. In the other direction, most of the spoken speech still sounds artificial and can be differentiated from a real speaking person. This gap is even larger when going beyond speech applications. Considering music, most of the automatic music transcription (AMT) systems perform worse than human experts and are mostly limited to percussive instruments like piano and drums [2]. This leaves out a bunch of more challenging pitched non-percussive instruments like bowed and wind instruments. Finally, there is all the research that focuses on non-speech – and non-music – audio which includes all sorts of sounds: animals, vehicles, noise, machines, etc. This latter category of audio signals makes the game harder and it may seem that high performances are hardly reachable, but we stick to the optimistic view of [3] stating that the processing of speech and music – and audio in general – by devices and the perception of these sounds by humans are areas that inherently interact with and enhance each other.

1.3 Audio analysis and applications

One way of organizing the ASP research topics is dividing them into four main groups: (1) *acoustic signal enhancement*: improving the sound's quality, (2)

acoustic modeling: modeling the interaction between sound and environment, (3) *audio reproduction*: creating perceptually relevant sound experience and (4) *audio analysis*: extracting information from sound. As it is clear from the thesis title, the focus of the presented work is on acoustic event detection (AED) which falls into the audio analysis (AA) group. Again, the increasing amount of digital audio nowadays resulted in thousands of AA tasks and techniques making it probably the most researched field of ASP today [4]. Speech recognition is perhaps the oldest topic in the AA field in which researchers aimed to transcribe a speech signal into text [5]. Other speech analysis topics were investigated like speaker identification – for security and dictation applications – and diarization [6] – identifying who is speaking at a certain moment. Similarly to the entire area of ASP, AA went discovering different types of audio, for instance, the rapidly growing research about the different music information retrieval (MIR) tasks [7] exploring the infinite world of music. In this field, the extraction of musical information ranges from extracting low-level features (loudness, timbre and pitch), deducing mid-level features (rhythm, harmony and melody), reaching out to high-level features (genre, similarity and emotions). The type of extracted information depends not only on the processed signal but also on the target application. Learning from the techniques used for speech – and music – analysis, AA extended its research to other types of audio giving birth to various topics like sound event detection, localization and acoustic scene classification.

Hopefully, the reader is convinced by now of the power of *audio*, and that it does not only traverse physical obstacles but also it reaches and can be integrated to thousands of our daily applications. In the following we introduce two AA research problems, for which this thesis will try to point out the different aspects and propose some answers to the related questions. Initially, it may seem that they only share the fact of being audio detection problems. Throughout the thesis, it will be clear that they share more than that.

1.3.1 Note onset detection

One research problem that is considered a building block for many of the MIR applications, is the note onset detection (NOD) problem. In this problem, researchers are interested in detecting, with high precision, the start of the different musical notes in a music excerpt. Those locations in time marking the beginning of notes are called onsets. It is easier to grasp the meaning of onsets by looking at an isolated note and specifically to its so-called Attack-Decay-Sustain-Release (ADSR) model. This model is extensively used by synthesizers for modeling acoustic instruments. It is modeling the evolution in the amplitude envelope of the played note [8] as shown in Fig. 1.1 for an accordion note. Its

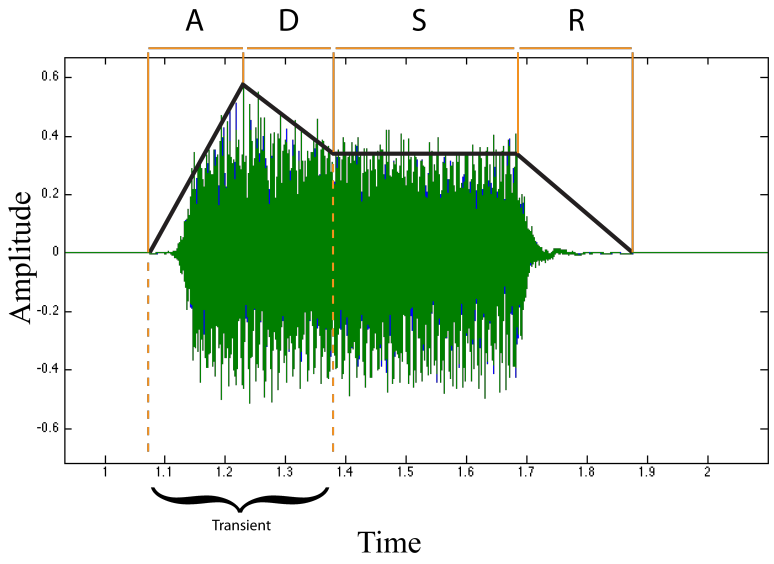


Figure 1.1: (A)ttack, (D)ecay, (S)ustain and (R)elease model (an isolated accordion note).

name comes from the abbreviation of the four model components described below:

- **Attack**: It represents how much time is needed by the signal amplitude (a played note) to reach its maximum value. This could be very short for some instruments especially the percussive ones, *e.g.*, percussion and guitar. Other instruments exhibit large attack time like bowed strings which belong to the pitched non-percussive family of instruments.
- **Decay**: This is the time to go from the highest reached level to the sustain level.
- **Sustain**: This is the time during which the note remains at a constant or steady-state level until it is released.
- **Release**: Finally this is the time during which the signal fades out from the sustain level until theoretically reaching the zero level.

More recent models for isolated notes suggested combining the spectro-temporal evolution with the amplitude envelope for better segmentation of the different

components of a note [9]. While the latter is important for handling the variability between the different instruments for the segmentation application, the ADSR is general enough for NOD as we are mainly concerned with the attack component which is assumed to always exist for all types of instruments and playing styles [9]. A simpler model divides the note into only two components in which a music note starts by a *transient* then a *steady-state* part. Similarly to different phenomena in life, the note starts with a shaky undecided state then reaches the zen state. More precisely, a transient is a short interval of time in which the signal energy and statistical properties change rapidly. It is the part of the signal representing the excitation and not the resonance. Figure 1.1 shows how the transient is the portion of the note where the energy content increases then decreases, *i.e.*, attack and decay. The human ear is able to distinguish between transients that are more than 10 ms apart [10]. Finally, the note onset is the chosen time instant to represent the note transient. It is chosen to be as close as possible to the start of the transient [11]. Hence the NOD problem could be formulated as looking for the transients of a music signal and trying to detect them as early as possible.

Enhancing the NOD performance still captures researchers' interest due to its role in many music processing and production applications. It is crucial for a better AMT [12]. Adaptive music effects make use of a so-called atomic description of music, separating transients from steady-state components of the notes [13]. For example, it is interesting to handle both components separately when applying the time-stretching effect. While the steady-state part of the note should undergo the stretching, the transient part should only be time-shifted but not stretched. Moreover, note onsets serve as features in many higher level applications like beat-tracking, tempo estimation, parametric coding and audio finger-printing for music search engines and intellectual property protection.

At first sight, when considering an isolated note, NOD may seem an easy task. In practice, the first challenge resides in defining where exactly an *onset* should be. In other words, when does a note start? Is it when the key is pressed or when the first sound is perceived? After that comes the annotation problem which is strictly connected to the performance evaluation. Answers to these questions are needed in the development of NOD algorithms capable of detecting all note onsets in a musical piece with a minimal number of false alarms. Those algorithms are expected to be generic, *i.e.*, capable of handling different types of instruments (pitched/unpitched, percussive/non-percussive) and playing styles (Jazz, blues, classic, etc.).

1.3.2 Howling detection

While for note onsets the target event is generally a pleasant sound, here the event is quite unpleasant and makes anyone run away. Howling is an audio artifact that is found usually in concerts or public address (PA) scenarios. It is this beep or whistle that occurs when the performer/speaker is positioning the microphone in front of a loudspeaker. This coupling between the microphone and the loudspeaker results in what is called acoustic feedback. This in turn creates a resonant behavior, causing the system to produce a howling sound. Usually, when it happens, the first and fastest solution is to lower the broadband PA gain to limit the annoying sound duration. This solution is quite poor especially if needed more than once, making a speech less intelligible or a music piece less enjoyable - by somehow hiding its dynamics. Several howling elimination solutions exist in literature which are inherited from the broader research problem of acoustic feedback control (AFC) [14]. The most interesting solutions are the ones that propose a narrowband gain reduction, manipulating the gain only in the surrounding of the resonant frequencies. This group of solutions is called notch-filter-based howling suppression (NHS), in which a howling detection (HD) algorithm aims to detect the howling components in order to activate proper notch filters to suppress howling [15]. As mentioned earlier, better performing HD algorithms can drastically boost the listener – and the performer – experience in PA systems. This also applies for hearing aids (HA) users where the distance between microphones and loudspeaker is small by default.

Like NOD, HD faces the same research questions related to dataset acquisition and annotation. Most of the available research uses either synthetic-howling datasets or real-howling recordings. In the former, howling is approximated by adding a sinusoid at a predetermined howling frequency to the source signal, which is far from realistic, while in the latter howling is intentionally produced and recorded for different room setups, but then the labeling afterwards by a human annotator is somehow subjective. Another big limitation of existing HD features, is that these are calculated only for a set of howling frequency candidates that are chosen by selecting a small set of spectral peaks. This limits the HD task to only check "how well a feature is able to differentiate howling from voiced speech or tonal music components for the selected set of candidates", totally excluding the case when howling has a lower magnitude than desired signal components. The use of existing HD features thus excludes the detection of ringing and early howling, which is however essential to suppress howling as early as possible. Moreover, since HD is an unbalanced event detection problem as there are generally much more non-howling realizations than howling realizations, the HD performance evaluation needs a more suitable metric than the ones that are used at the moment.

1.4 Contributions

After having introduced the two AED tasks, NOD and HD, and having detailed the different challenges for each, in this section we point out the different contributions of this work, in response to those challenges. The contributions for both tasks, consist of improving datasets, proposing new features, and redesigning evaluation procedures. These contributions are organized in different chapters as follows:

- Chapter 3: a new way of generating annotated datasets for NOD, inspired from the onset definition given by [16] where onsets are the first detectable part of a musical note in an isolated recording. A novel "Mix-Notes" MATLAB tool has been developed which first annotates isolated notes and then mixes them, hence generating automatically annotated excerpts. This kind of datasets can be generated with less human effort, in large quantities, and with the property of being human and note-context independent. This latter property removes any dependency between training/tuning and testing datasets.
- Chapter 4: normalised identification of note onset based on spectral sparsity (NINOS^2), a new feature for non-data-driven NOD based on a spectral sparsity measure in addition to signal energy. We also propose an atypical pre-processing step making the feature totally dependent on the low-energy frequency components in a signal frame, traditionally neglected but proven here to be tightly connected to onsets. Both contributions, *i.e.*, pre-processing and feature design, significantly improve the NOD performance for the challenging sustained-strings instrument class.
- Chapter 5: a study about the impact of semi-synthetic data augmentation on data-driven NOD. An NOD experiment is conducted on piano datasets using three augmentation strategies. Preliminary results show a slightly better and more stable, *i.e.*, lower-variance, detection performance.
- Chapter 6: introduction of a new and crucial hyperparameter "*annotations time shift*" for training/evaluation of data-driven NOD models using differently annotated data sets (manual, automatic, MIDI).
- Chapter 7: a large annotated dataset for HD in which howling is simulated with a closed-loop feedback system. The dataset is generated using different acoustic impulse responses (AIRs) covering a wide range of howling frequencies. It was generated for both speech and music input signals. The dataset is more realistic than simulating howling with sinusoidal superposition, and annotations are generated automatically, hence human-independent.

- Chapter 8: NINOS^2-Transposed (NINOS^2-T), a novel feature for HD based on spectral sparsity. This feature is applied directly on the full spectrum, considering all frequency-time bins as howling candidates. This overcomes the limitations of the candidate-selection method used in earlier research and opens the door for complete howling detection. Moreover it makes it possible to perform ringing and early howling detection. A second contribution of this chapter consists in improving the traditional HD evaluation by replacing receiver operating characteristic (ROC) visualization and the area under the ROC curve (ROC-AUC) metric with the well-known precision-recall (PR) visualization and the area under the PR curve (PR-AUC) metric. The ROC-based evaluation is shown to be unable to correctly assess the performance for tasks with highly imbalanced datasets such as HD. Pushing the HD assessment further, two evaluation modalities are suggested: early and full detection.

1.5 Outline of the Thesis

The remainder of this thesis is divided into four parts. In Part I, having introduced the research area and the applications of interest in Chapter 1, Chapter 2 will introduce the specific type of events studied in this thesis and how this links the two applications, NOD and HD. It also describes a detection methodology for this type of events together with an overview of the state-of-the-art solutions for each of its steps. Finally, Chapter 2 is terminated by confronting different evaluation methods and metrics highlighting the pros and cons of each.

Part II focuses on the NOD problem tackling the challenges pointed out in Section 1.3.1. In response to the different NOD dataset limitations, Chapter 3 and Chapter 5 suggest the usage of semi-synthetic datasets as a solution. While Chapter 3 introduces the Mix-Notes tool for automatically annotated dataset generation, Chapter 5 studies the effect of using this kind of dataset for the training of a Deep Neural Network (DNN) for onset detection. Chapter 5 is based on the publication:

- M. Mounir, P. Karsmakers, and T. van Waterschoot, "CNN-based Note Onset Detection using Synthetic Data Augmentation," in Proc. 28th European Signal Process. Conf. (EUSIPCO '20), Amsterdam, Netherlands, Aug. 2020.

Chapter 4 introduces NINOS^2, a novel feature for non-data-driven NOD and it is based on the publications:

- M. Mounir, P. Karsmakers, and T. van Waterschoot, "Guitar note onset detection based on a spectral sparsity measure," in Proc. 24th European Signal Process. Conf. (EUSIPCO '16), Aug. 2016, pp. 978–982 (*best paper award*).
- M. Mounir, P. Karsmakers, and T. van Waterschoot, "Musical note onset detection based on a spectral sparsity measure," submitted for publication.

Finally, a crucial hyperparameter named "annotations time shift" is suggested in Chapter 6, allowing for a more fair NOD performance comparison. Chapter 6 is based on the publication:

- M. Mounir, P. Karsmakers, and T. van Waterschoot, "Annotations Time Shift: A Key Parameter in Evaluating Musical Note Onset Detection Algorithms," in *Proc. 2019 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '19)*, New Paltz, NY, USA, Oct. 2019, pp. 21–25.

Then in Part III, the focus moves to the HD problem. Chapter 7 details the generation of an improved howling-corrupted dataset that is automatically annotated. In Chapter 8, a novel HD feature NINOS^2-T is introduced and its performance is compared to the state-of-the-art features. Chapter 8 is based on the publication:

- M. Mounir, G. Bernardi, and T. van Waterschoot, "Robust and early howling detection based on a sparsity measure," submitted for publication.

Finally the conclusions are presented in Chapter 9 together with suggestions for future work.

Chapter 2

Acoustic event detection

Reviewing the acoustic event detection (AED) literature, one can easily recognize it appearing under different names exchanging its first word with audio, sound, audio-based or even more specifically non-speech audio. This comes as a consequence of how AED was first introduced to handle all types of sounds other than speech, since speech analysis was already a more established research problem back then. Recently, AED was defined in [17] as one type of problems encountered in the wider field of Computational Analysis of Sound Scenes and Events (CASSE). They described the AED task as to locate in time the occurrences of a specific type of sound or sounds either by finding:

- each instance of the sound(s). In other words, the AED algorithm searches for the target sound event.
- all the temporal positions at which the sound(s) are active. Here, the AED algorithm produces some sort of activation function determining the existence - and non-existence - of the target event at every time instant.

This task should be differentiated from the *classification* task in which the goal is to categorize an audio recording into one category of a set of predefined categories [17]. It has been hypothesized that classification is a simpler task than detection as the discrimination is principally between events known to the classifier while in detection, the algorithm is trying to discriminate the target event from whatever other sound, *i.e.*, form various events and from highly rich background sounds [18].

Speaking about AED becomes clearer and more complete by considering the targeted event(s) which follows from the target application. For instance, to cope

with the enormous amount of existing digital audio which keeps growing, there is a continuous need for context-based indexing and retrieval applications [19, 20]. Health monitoring [21] and analysis of elderly daily living activities [22] are also growing research fields exploiting AED techniques due to the unobtrusive property of audio compared to video, ensuring more patient privacy. In those applications researchers are interested in detecting events like collapsing, falling down stairs, conversation, walking and screaming. All those applications are hot topics that keep attracting researchers for a better performance quest as it is a quality-of-life changer. Moreover, applications for AED will keep popping up, as their is no limit of what could be inferred from audio: human actions (health, security), animal voices (diseases, welfare), machine sounds (faults detection), and so on. The story does not stop at detection but continues to inference based on the detected events, which gives rise to tons of higher-level applications, for instance: scene classification [17], emotion recognition [23], smart rooms [24] and smart environments in general.

2.1 Acoustic Events

Considering the applications under study, out of the different naming for AED we prefer using the word *acoustic* rather than *audio*. Using the word *audio* may be misleading in our study by limiting the scope to favorable audio for human perception like human speech, animal sounds or musical instrument notes. The word *acoustic* represents a wider scope, including not only speech and music, but also any other sound signal even if it is an undesired sound, *e.g.*, the howling artifact.

But what is an event? Thinking out loud, events are those things without which our life would be extremely boring. In [25], events are defined as "things that happen". In reality, everything that happens as a function of time can be described by a single or a sequence of event(s). "Everything" could be singing a song, biking 20 km, saying "hi" or even thinking of an apple. Here we are interested in a subgroup of events that can be perceived by humans and specifically the ones resulting in a perceived sound. This notion of event is of great importance for a computer scientist who needs to translate human to machine perception. It serves as the abstraction dictionary between both worlds summarizing what happens in the real world with fewer digital tags.

More concretely, an acoustic event refers to a specific sound produced by a distinct physical sound source for a precise amount of time [17]. As in the audio signal processing (ASP) field, where audio signals are often analyzed in both time and frequency domains, we may redefine the acoustic event being limited

not only to a certain period of time, but also to a specific frequency region. This thesis focuses on acoustic events that have distinctive time-frequency patterns that differentiate them from other events – including background ones. These patterns are, under certain conditions, visible when analyzing the source signal's spectrogram – a time-frequency representation of the source signal. In practice, AED in this context is not an evident task and the need for powerful detection algorithms is crucial to uncover these specific patterns also when hidden or noisy. For this type of AED problems, the difficulty of finding a solution is dependent on two main aspects:

1. Intra-class variations: how similar are different instances of the same event.
2. Inter-class variations: how dissimilar are instances belonging to different events.

2.2 Brief history of AED research

Most of the existing solutions for AED consist of two components:

- Seeking **features** that accentuate the aforementioned similarity and dissimilarity aspects. Practically, features provide a different representation of the source signal in which the target events are highlighted. Finding an adequate representation is accompanied with – or comes as a result of – reflecting on the discriminating characteristics of the event in different domains like the time domain (temporal), the frequency domain (spectral) or a mixture of both.
- The features are usually designed in parallel with a **detection algorithm**. The latter is required for pointing out the time locations of the events either as an activation function or as a list of onset/offset times.

As solutions to AED were initially inspired from the ASP field, the first solutions [26, 27] used local features calculated for short time frames to train a Gaussian Mixture Model (GMM). On top of that, a Hidden-Markov model (HMM) is used to localize the events in time. Alternatively, in [28], so-called i-vectors were used to represent an entire audio stream with a relatively low dimensional feature vector retaining the relevant statistical information modeling only the changes related to the considered acoustic event – and channel – with respect to a pre-estimated Universal Background model (UBM). The UBM is a more global model summarizing information for longer frames and more event instances.

The i-vectors were used to train a Support Vector Machine (SVM) binary classifier per event. More recent work suggests the use of dictionary-based solutions. For instance, in [29], Mel-Frequency Cepstrum Coefficients (MFCC) features are adopted to learn a dictionary using a convolutive Non-Negative Matrix Factorisation (NMF) model. In this method each event is represented by non-negative atom(s) selected from a dictionary using an activation function. Another exemplar-based NMF was proposed in [30] were the atoms are pre-sampled – and augmented – from the spectrograms of the different events defined in a training set. Next to these algorithms, there are the Neural Network (NN) solutions and more specifically the Deep Neural Network (DNN), that are often outperforming more traditional solutions. In [31], the best performing NN approaches for AED were reviewed, including but not limited to the Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Convolutional Recurrent Neural Network (CRNN). These NN models were trained and tested on datasets containing a variety of (sound) events like people walking/speaking, car sounds and children playing. The best performer in terms of error rate and F_1-score (F_1) was the CRNN model.

2.3 Note onsets and Howling: Spectrally connected

Considering the two alternative definitions given for AED, it is clear that note onset detection (NOD) is looking for the time location of notes in a music excerpt while howling detection (HD) should yield some kind of activation function signaling the presence of howling. The first detection problem focuses on appealing sounds forming a melody while the second problem relates to seeking repelling sounds - *i.e.*, artifacts - degrading both the performer and the listener experience. While NOD is restricted to music signals, HD is applicable whenever there is a possible acoustic coupling between a microphone and a (loud)speaker, hence for both music and speech signals.

At first sight, both audio events seem to have less to share than to distinguish. Nevertheless, from a detection perspective, they share almost the same time-frequency pattern. Even though note onsets and howling bear different meanings, it suffices to rotate one of their respective spectrograms by 90° and the resemblance between their patterns is clear. This is illustrated in Fig. 2.1 showing a music excerpt's spectrogram in which the two types of events are present: a howling artifact around 400 Hz marked with an horizontal ellipse and several notes for which we mark the onsets of three examples happening at 9.5 sec, 12.1 sec and 16.9 sec. One can easily notice the common time-frequency pattern, which is a vertical line for the onsets and a horizontal one for the howling. Restating this from a spectral sparsity point of view, both events

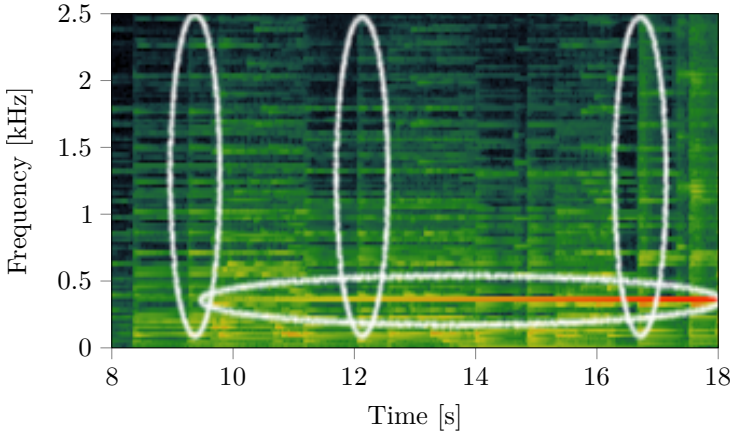


Figure 2.1: A music excerpt's spectrogram identifying similarity between note onsets (vertical ellipses) and howling examples (horizontal ellipse).

can be somehow distinguished being less sparse compared to their surrounding (*i.e.*, neighboring time instants for note onsets or neighboring frequency bins for howling), with spectral sparsity being a measure of the energy distribution among vertically or horizontally aligned time-frequency bins giving higher scores for when energy is concentrated in fewer number of bins than when uniformly distributed across all bins. Spectrograms show a lower spectral sparsity along the frequency axis for note onsets while the low sparsity appears along the time axis for the howling phenomena. Spectral sparsity is the key novel idea behind the detection algorithms developed in this thesis. It is the feature connecting onsets and howling as it cares less about energy magnitude and reflects the energy spread. In other words, even though howling amplitude increases over time while onset energy is stable across the frequency axis, both phenomena are alike from a spectral sparsity point of view. Spectral sparsity will be revisited in more detail, including its adapted use for each of the applications, in Chapters 4 and 8.

2.4 General AED scheme

Figure 2.2 shows a general solution scheme for the AED problem that is adopted in several NOD [11] and HD [15] algorithms. This three-step solution scheme is suitable for the type of events under consideration here, for which the detection algorithm is supposed to find specific time-frequency patterns in a given source

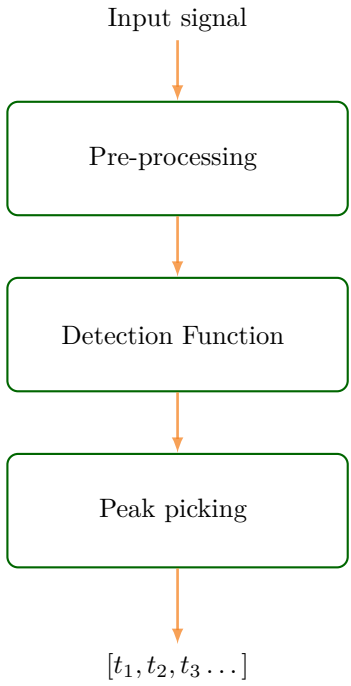


Figure 2.2: Solution scheme for AED.

signal (referred to as *input* signal in the sequel). The solution scheme starts by a pre-processing step which aims to find the most suitable input signal representation to detect specific patterns in the next steps. This could consist of filtering out information irrelevant to the target event. Alternatively, it could be a simple transformation into a different domain in which the events are highlighted. The second step of the solution scheme in Figure 2.2 is the pattern seeking step, named after its output which is the *detection function (DF)*. The DF is expected to have fewer (or sometimes the same number of) time-frequency samples as the original spectrogram but most importantly, exhibiting an energy rise at target event locations in time and/or frequency. In other words, it highlights the time instants or more generally the time-frequency bins corresponding to target events. The DF can be found in literature under different names like *reduction function* or *novelty function*. The notion of reduction refers to its down-sampling capability, whereas novelty refers to the occurrence of statistically novel information and is tightly related to the use of a DF for NOD considering every note as something new that is happening. The novelty interpretation of note onsets explains how music – like

what was attributed earlier to life – is an event-based phenomenon from both the performer and listener point of view [11]. The final processing step called peak picking (PP) is a further reduction function that aims to estimate precise location(s) in time and/or frequency at which the event is occurring. In its most general form, it is some kind of local maxima detector.

In the following, we will review the different approaches proposed in literature for each of the steps in the solution scheme while trying to find analogies to the general AED solutions and to focus on the two applications studied in this thesis.

2.4.1 Pre-processing

This initial step can be thought of as the guide for the feature seeking part of the AED solution introduced in Section 2.2. It is supposed to discard information that is irrelevant to the target application and hence unnecessary for the subsequent DF step. On the other hand it can serve to accentuate the event's relevant features. For instance it could simply be a transformation into another domain in which the target events are highlighted. The most commonly used transformation for audio processing is the transformation into the frequency domain. This can be done using the short-time Fourier transform (STFT), generally grouped – for successive time frames – in a spectrogram, as proposed in [15] for HD and [32, 33] for NOD. Alternatively, various AED applications make use of a more psychoacoustically aware frequency transformation, namely the Mel-based transformations like MFCC, Mel-band energy and Mel spectrogram [29–31, 34]. In ASP, grouping spectral features is not limited to magnitude spectrograms but could also be done by calculating feature supervectors or i-vectors that are more convenient for longer events [28]. All of those transformations may be preceded by the application of a windowing function – to reduce spectral smearing and sidelobe interference – depending on the application. In some cases, it is beneficial to use a more elaborate pre-processing than just a domain transformation. Here are a few of the more advanced pre-processing techniques used in AED literature:

1. **Frequency Sub-Band Processing**
   This technique divides the input signal into frequency sub-bands and applies the pre-processing – and in some cases also the DF step – to each sub-band separately, after which the final output is obtained as a combination of the different sub-band results. Firstly, this increases the robustness of the detection function by relying on many (sub-band) opinions like in a voting system. Secondly, it will probably increase the visibility of events as these generally appear differently in the different

frequency bands. For instance, using constant-Q filters and assessing energy changes for higher frequencies and spectral changes for lower frequencies enhanced the NOD performance in [35]. The same technique was used for the NOD approach proposed in [36] and for the beat and tempo tracking proposed in [37] to inject psychoacoustic knowledge into the AED solution. Similarly for HD a solution using the Generalised Teager-Kaiser Operator (GTKO) with a different parameter per frequency band is proposed in [38].

2. **Transient/Steady-State Separation**
   This pre-processing technique follows from the ADSR model for music signals shown in Fig. 1.1 and is mainly useful for the NOD problem. It aims to differentiate transients from steady-state components, often starting from a sinusoidal model of the input signal. A first technique proposed by [39] is the Spectral Modeling Synthesis (SMS) which explicitly models the tonal – *i.e.*, steady-state – component of a musical note, and then calculates the transient from the residual of the input signal after removing the tonal part. In a second technique, Transient Modeling Synthesis (TMS) [13], the transients are explicitly modeled to avoid considering that noise in the residual of the SMS technique is considered to be part of the transient.

3. **Voice Activity Detection (VAD)**
   From its name, this technique is suitable for detecting speech activity but it can be extended to general AED by using some general activity detector. For instance this was used by the HD solution suggested in [40], to first separate speech from non-speech parts of the signal and then apply howling detection and suppression. While this does not trivially solve the HD problem as howling may coexist with speech, it avoids affecting the speech signal when it is still intelligible.

2.4.2 Detection Function (DF)

This is by far the most important and challenging step of the AED solution scheme as, differently from the pre-processing step that typically involves general ASP techniques, the DF step often requires a deeper domain knowledge and is customized per application. Moreover, designing a good DF saves us the effort of finding sophisticated peak-picking algorithms as simple local maxima detectors would do the job. For that reason, it is not surprising that the DF design problem attracts most research interest when trying to improve the overall AED performance. Coming up with novel DFs is a significant part of

the presented thesis works. It has also formed the starting point for other contributions of this thesis in dataset design and performance evaluation.

In terms of the AED solution introduced in Section 2.2, even though DF starts with the *detection* keyword, the design of a suitable DF actually corresponds to the feature-seeking step rather than the detection algorithm step. When introduced for the NOD problem, the DF was defined as a highly sub-sampled version of the input signal emphasizing its target events [11]. Considering music as a sequence of transitions, the reduction function was conceived to detect specific signal changes [41], more precisely, to detect onsets while omitting other types of changes like offsets, noise bursts and amplitude modulations (*e.g.*, vibrato and tremolo effects). Usually, the note onset DF is a 1D time domain signal having fewer time samples than the original input and exhibiting an amplitude increase at onset locations. Moving to the HD problem, a 2D DF is usually calculated from the input spectrogram, *i.e.*, maintaining both the time and frequency dimensions. In Part III, it will be explained how the two DF properties, *i.e.*, highlighting and reduction, still apply for the HD problem. For other AED tasks, the DF design may encompass both feature-seeking and detection into one algorithm with no need for a subsequent PP step.

After grasping the general idea of what a DF is expected to show, in the following we go over the different DFs proposed in literature highlighting their strengths, weaknesses and possible improvements. This will be done in a way to follow how historically the DF was introduced for the NOD problem with references to the state-of-the-art DFs. Moreover we briefly discuss the state-of-the-art methods for HD and more generally for AED that are analogous to this step of the solution scheme. For this we will adopt the DF grouping introduced in [11], and expand it to consider DFs for the HD and other AED tasks:

- **Temporal features**
  As audio signals are naturally represented in the time domain after digitization, the first DFs were designed to track the temporal evolution of the input signal. For instance the *envelope follower* was suggested in [42] for NOD and is given by:

$$E(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} |x(n+m)| w(m), \tag{2.1}$$

  with $x(n)$ being the signal in the time domain and n the discrete time index. From its name, this DF expects an amplitude increase accompanying each and every event present in the signal. It is calculated per frame of length N. Each frame is first rectified and then smoothed using a window $w(m)$ of the same length. This DF is simple and computationally

inexpensive but it performs well only when the signal exhibits a distinctive and fast magnitude increase at event locations, as is the case for percussive musical instruments such as drums, piano and guitar. Contrarily, it fails in detecting onsets for non-percussive instruments like violin in which the transient may be too slow and expanding over many time frames. In a variation of this DF, the signal samples are squared instead of rectified, resulting in a *signal energy follower*. On top of that, researchers have suggested the use of the time derivative of the energy or of its log-magnitude in order to avoid spurious onsets due to amplitude modulations. This was suggested together with a sub-band pre-processing in [37] and with a transient/steady-state separation in [43]. These variations were shown to improve the detection performance for percussive instruments but still failed with the non-percussive ones.

- **Spectral features**
  In many ASP applications, the input signal is analyzed in the frequency domain. This shares the same motivation as the sub-band pre-processing approach described earlier. Both approaches exploit the fact that audio events are characterized by different energy distributions among the different frequency (sub-)bands and hence can be analyzed by studying their respective time-frequency patterns. For instance the high-frequency content (HFC) DF suggested in [44] for NOD is based on the fact that transients – in contrast to the steady-state portion of a music note – exhibit more pronounced changes in the input signal statistics in a short time, resulting in an increase of high-frequency signal energy. Hence for emphasizing high-frequency content, the HFC-DF linearly weights the frequency (sub-)band contents proportionally to their frequency,

$$HFC(n) \;=\; \frac{1}{N}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} W_k |X_k\,(n)|^2, \tag{2.2}$$

  with k denoting the frequency index, $X_k(n)$ the STFT of the windowed input signal $x\,(n)$,

$$X_k(n) \;=\; \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} w(m)x(n+m)e^{-\frac{2j\pi mk}{N}}, \tag{2.3}$$

  and $W_k = |k|$ is the HFC weight for frequency bin k.

  A different family of spectral features monitors magnitude changes in different frequency bins, expecting a higher amount of change to be linked to input signal transients. To detect onsets, these methods average the

change of spectral magnitude or spectral dissimilarity across the different frequency bins. This idea of Spectral Flux (SpecF) was introduced for NOD by [44] using an ℓ_1-norm distance metric and modified into an ℓ_2-norm distance metric in [35],

$$SpecF(n) \;=\; \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \left\{H\left(|X_k\left(n\right)| - |X_k\left(n-1\right)|\right)\right\}^2, \tag{2.4}$$

with H being the half-wave rectification operator keeping track only of positive changes and neglecting the negative ones representing offsets,

$$H\left(x\right) \;=\; \frac{\left(x+|x|\right)}{2}. \tag{2.5}$$

Two more variations of the flux family were used with NOD. The first, Logarithmic Spectral Flux (LSF) achieved a better performance by using the log-magnitude of the spectrum [45]. The second is Super Flux (SF) [32] which has reported the highest score in the Music Information Retrieval Evaluation eXchange (MIREX) competition for several years. The latter employs a maximum filter technique for calculating the spectral difference to counter the spurious onsets due to vibrato and tremolo effects. Nevertheless, even if these methods are simple and computationally efficient, they fail in detecting onsets when the successive notes share many harmonics without a sufficient increase in magnitude during the note onset. This happens regularly for chords and repeated notes.

In contrast to the spectral DFs described above which are based on the magnitude spectrum only, other approaches have suggested to consider also the phase spectrum as it contains a large part of the signal's temporal structure [11]. In [46], the non-peakiness of the phase distribution was used to design an onset DF as this distribution tends to be wide and flat for transients in contrast to the majority of the steady-state portion having an almost constant phase. In [47], a simpler measure representing the same phenomenon, namely the mean absolute phase deviation, was proposed. Two enhancements were suggested in [41], weighting each frequency bins' phase deviation by its respective magnitude to give more importance to the sounding partials, and normalizing the weighting operation resulting in the Normalized Weighted Phase Deviation (NWPD) DF,

$$NWPD(n) \;=\; \frac{\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |X_k\left(n\right)\varphi''_k\left(n\right)|}{\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |X_k\left(n\right)|}, \tag{2.6}$$

where φ'' represents the finite-difference change in the instantaneous frequency

$$\varphi''_k(n) = \varphi_k(n) - 2\varphi_k(n-1) + \varphi_k(n-2), \tag{2.7}$$

and φ is the 2π-unwrapped phase. Unfortunately, the phase-based DFs are vulnerable to low energy noise and distortion, and are therefore usually used in combination with other DFs.

Alternatively, a DF that combines the magnitude and phase spectrum is the Complex Flux (CF) [11, 41]. Again, this measure expects a constant magnitude and phase deviation when no transients exist. It is computed by defining the predicted complex magnitude spectrum $\widehat{X_k}(n)$ as

$$\widehat{X_k}(n) = X_k(n-1)e^{\varphi_k(n-1)+\varphi'_k(n-1)} \tag{2.8}$$

with $\varphi'_k(n-1)$ denoting the phase difference between the two previous time instants, and hence defining the CF as

$$CF(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \left| X_k(n) - \widehat{X_k}(n) \right|. \tag{2.9}$$

Not surprisingly, the CF and its rectified version Rectified Complex Flux (RCF) presented in [41] are sharing many properties with the SF-DF as they all come from the spectral flux family of DFs.

Similarly for the HD problem, different spectral features were developed to capture specific howling properties. In the reported state-of-the-art methods [15], few howling frequency candidates are selected based on their energy content, then a howling feature or DF is calculated pointing out the subset of candidates that most probably correspond to howling frequencies. For instance, because the howling frequency power is usually higher compared to other frequencies, the Peak-to-Threshold Power Ratio (PTPR) and Peak-to-Average Power Ratio (PAPR) features were suggested to detect a howling frequency whenever its corresponding power surpasses a certain threshold. Other DFs like Peak-to-Harmonic Power Ratio (PHPR) and Peak-to-Neighboring Power Ratio (PNPR) compare the candidates' power to the power of their harmonics or neighbors. This is because howling is expected to be a purely sinusoidal component, hence the howling component power is concentrated in a single or a few frequency bins and with no harmonics. Another measure, Interframe Peak Magnitude Persistence (IPMP), used the persistence of a howling frequency candidate over time as an indication for howling. A more sophisticated measure, Interframe Magnitude Slope Deviation (IMSD),

exploits the fast magnitude increase for howling with respect to other signal components. All of these measures were designed to identify howling from non-howling frequencies out of a small set of pre-selected candidates. This is the main limitation of these methods as they are not able to detect howling in early stages – *i.e.*, when it is not audible – since the candidate selection will not include an early howling component when it is still lower than the source signal components or when it is just a ringing effect. This limitation comes on top of the individual disadvantages that exist for each of the aforementioned features, such as their high sensitivity to threshold variation, in particular for the IMSD feature for instance. The mathematical definition for each of these measures and a deeper analysis of their performance in comparison to newly proposed features is detailed in Part III.

- **Data-driven DFs**
  A totally different approach for DF design, relies on learning a DF from annotated data rather than analyzing the event properties and handcrafting suitable features. This approach started from the early solutions to AED using the combination of GMM and HMM [26,27]. There are also the dictionary-based approaches using convolutional-NMF [29] or exemplar-based NMF [30] to learn atoms that represent the target events. Alternatively an SVM binary classifier was trained per acoustic event in [28]. Similarly an SVM was proposed in [48] for detecting the presence of howling but without estimating its frequency. More recently, DNN approaches have been proposed that often outperform traditional AED methods, for instance the RNN [49] and CRNN [50] proposed for AED in real-life polyphonic recordings. In most – if not all – of the aforementioned AED data-driven methods, several events are allowed to coexist and the output probabilities are not assumed to sum to one. In the context of the presented AED solution scheme, this can be seen as a separate DF for each of the target events. On the other hand, NOD DNNs are often trained on binary labels treating the onsets as the only existing event and the rest of the signal as no event, hence obtaining the DF as the network output. When the resulting DF is limited between 0 and 1, it can be interpreted as the onset probability giving rise to the classification of NOD solutions into probabilistic and non-probabilistic models [34]. This applies for the Bidirectional Long Short-Term Memory (BLSTM) RNN proposed in [51], and it is also the method used by the state-of-the-art data-driven NOD method using a CNN [34]. In general, these data-driven methods need huge amount of annotated data which is usually the bottleneck for generalization to unseen datasets. In the sequel we keep the more general classification of NOD-DFs into data-driven and non-data-driven (temporal and spectral) solutions.

Recent work suggests using an end-to-end approach for the data-driven models. This recent technique replaces the traditional two-step model which separates feature-learning from classification with a simultaneous end-to-end training for both components [52]. In a certain way, it shrinks or replaces the different steps of the suggested AED solution scheme in Fig. 2.2. For instance, the automatic music transcription (AMT) proposed in [53] for piano recordings, used a NN acoustic model followed by a RNN music language model and trained them together. The acoustic model learns pitch probability in an audio frame while the music language model learns the combination of pitches over time. Nevertheless, the authors retained the pre-processing step as the input to the network consisted of commonly used features and not the raw audio as in [52]. In the work presented in this thesis we preferred separating the three steps as, on one hand, they still match the human approach for auditory scene analysis [54]. On the other hand, the end-to-end approach can still benefit from this scheme for its design and analysis. For the design, it helps in choosing the model structure while for the analysis, it can help getting insights from the learned representations in the hidden layers.

We limited our review to the standalone DFs but there are many works in which a fusion between different DFs is considered, such as the method proposed in [55] for NOD and the different feature combinations suggested in [15] for HD.

As the DF is the core of the AED solution scheme, we want to emphasize how difficult it is to obtain a fair and accurate comparative performance evaluation for the different DFs proposed in literature. For instance for NOD, a review paper [11] reports HFC and negative likelihood as the best performing DFs while another [41] suggests the SF and CF. This can be due to several reasons: tuning of algorithm hyperparameters, choice of datasets, dataset annotation method and dataset division into train/test sets. For comparison purposes and baseline performance, most researchers refer to yearly competitions like MIREX. For instance the MIREX results report CNN [34] and CF [56] as the best performing data-driven and non-data-driven algorithms respectively.

2.4.3 Peak Picking (PP)

Here we discuss the third and final step in the AED solution scheme, that performs the actual event detection and whose complexity largely depends on the previous reduction step and how representative the DF is of the target event(s). In some methods, the PP step is contained in the DF step, which is the case for example for binary classifiers that directly output the predicted classes (event vs. no event). For high-quality DFs, the PP step boils down

to a local maxima search. In general, the PP step can be divided into three substeps:

1. **Post-Processing:** This is a "polishing" substep applied to the DF before thresholding it. For instance, it could be a low-pass filter smoothing the DF, retaining only the peaks that could be event candidates. In some applications that do not require online processing – *i.e.*, detection in real-time – the DF undergoes normalization and DC removal in order to limit the threshold search space.

2. **Thresholding:** From its name, this substep separates DF peaks that represent events from irrelevant DF peaks. It could be *Fixed Thresholding*: comparing the DF with a constant threshold. This form of thresholding is the simplest and may fail with signals having stronger dynamics like quiet and loud passages in music excerpts. Nevertheless it is used with the state-of-the-art data-driven CNN by [34] for NOD as those deep classifiers generally output very precise DFs that are quite peaky around the onset locations. This is favorable for the thresholding but on the other hand these DFs are more susceptible to other problems such as the annotations misalignment described in more detail in Chapter 6. Alternatively, *Adaptive Thresholding* is used, especially with non-data-driven – handcrafted – DFs. Usually, the threshold is adapted to the overall amplitude or power variations of the input signal.

3. **Peak Picking:** Finally this substep that holds the same name as the entire PP step, selects a specific point in time to represent each of the DF peaks that are kept after thresholding. For the NOD problem for instance, it outputs a list of time stamps where note onsets are detected. A simple PP implementation would be a local maxima detector. In practice, more customized PP algorithms are used depending on the target event(s). For instance, with the state-of-the-art non-data-driven NOD method [32], the three PP conditions – introduced earlier in [41] – were refined for considering a point on the DF as an onset:

   (i) an onset is the highest-amplitude DF point in its neighborhood,
   (ii) an onset has a DF amplitude higher than an adaptive threshold calculated as a long-term average of the DF,
   (iii) two onsets cannot occur closer than a certain time apart.

   More details about this PP method and its pros/cons will be explained in later chapters, as it will be the one used in the presented work.

A final remark about PP is that in many research articles, researchers are less concerned about coming up with optimal threshold value(s) as this is

highly application-dependent. They do instead care about assessing how well on average an AED method performs and how sensitive it is to threshold changes by small to large offsets. It is clear the choice of thresholding methods and threshold values are tightly related to performance evaluation which will be introduced in the next section.

2.5 Performance Evaluation

Last but not least, we conclude this chapter by discussing different performance metrics and evaluation methods proposed for AED. In fact, even though this topic was left to the end for a better understanding of how it applies to the problems under study, the evaluation is as important as the solution itself and is usually designed at the start of the research process. In general, an AED algorithm needs to be evaluated on its capability to detect the events without being misled by surrounding nuisances. In the two applications studied here, there is only one event to distinguish for each: onsets and howling. Hence, both tasks can be treated as binary classification (BC) problems. The training and evaluation datasets, in that case, are labeled with (+)'s for the locations marking a positive realization of the target event while the remaining locations are labeled with (−)'s. In other words, only the time frames that hold a note onset and the time-frequency bins corresponding to the howling frequency are marked with (+).

2.5.1 Traditional binary classification evaluation metrics

The different evaluation metrics proposed for classification problems follow from what is called the *confusion matrix* – or *contingency table* – shown for a BC problem in Table 2.1. It summarizes the relation between the ground-truth annotations (columns) and the detection labels (rows) using four values described below:

True positive (TP): the number of positive instances that are correctly classified.

False positive (FP): the number of negatives that are wrongly detected as positives. These are frequently called *false alarms*.

True negative (TN): The number of negatives that are correctly classified.

False negative (FN): the number of missed positives wrongly classified as negatives.

Table 2.1: Confusion matrix for event detection problems.

		Ground Truth (+)	**Ground Truth** (−)	**total**
Prediction Outcome	(+)	True Positive	False Positive	D_+
	(−)	False Negative	True Negative	D_-
total		GT_+	GT_-	

On the table margins, the column totals, GT_+ and GT_-, represent the total number of events and event-free instances present in the signal. On the other hand, the row totals, D_+ and D_-, represent the total number of instances detected as positive and negative, respectively. Finally, the total number of test points – time frames for NOD and time-frequency bins for HD – is equal to $(GT_+ + GT_-) = (D_+ + D_-)$. Based on these values, several evaluation metrics have been developed. We limit ourselves to the five most used ones, each trying to reply to a specific question:

Accuracy (Acc): *How well does the algorithm/feature classify both positives and negatives?* Acc assess equally the classified instances regardless of their class [57].

$$Acc = \frac{TP + TN}{GT_+ + GT_-} \tag{2.10}$$

True positives rate (TPR): *Out of all the GT_+ positive instances, what is the percentage of detected ones?* This measure is also known under the name Recall (R).

$$TPR = R = \frac{TP}{GT_+} \tag{2.11}$$

False positives rate (FPR): *Out of all the GT_- negative instances, how much are wrongly detected as positive?*

$$FPR = \frac{FP}{GT_-} \tag{2.12}$$

Precision (P): *Out of all positive detections, what is the percentage of correctly classified ones?* This measure differs from TPR in its reference, as it compares to the D_+ positive detections instead of the GT_+ positive ground-truth instances.

$$P = \frac{TP}{D_+} \tag{2.13}$$

F_1-score (F_1): This metric is defined as the harmonic average of P and R. It is good in averaging ratios and mitigating the effects of outliers. In contrast to arithmetic means, a harmonic mean – defined as the reciprocal of the arithmethic mean of its components' reciprocals – tends strongly towards the least of its components.

$$F_1 = \frac{2PR}{P+R} \tag{2.14}$$

All these single-valued metrics share the property that they are threshold-dependent, *i.e.*, they are calculated for a specific detection threshold θ. Usually in the evaluation process, the dataset is first divided into training and testing sets. The threshold $\theta^\star$ maximizing the metric on the training set is then used to calculate the metric estimate on the testing set. This makes these metrics quite dependent on the way the dataset is collected/designed and divided into training and testing sets. Although these metrics allow for an easy performance comparison due to their single-valued nature, they do not show how the algorithms would perform if the threshold $\theta^\star$ were altered.

2.5.2 Graph-based evaluation

Alternatively, to achieve a more elaborate assessment, researchers have used graph-based evaluations by defining plots of different metrics versus each other. Due to its visual nature, this evaluation approach makes the assessment much more convenient. Moreover, it allows to compare algorithms independently of their detection thresholds. The most widely used is the receiver operating characteristic (ROC) curve, which plots TPR versus FPR, calculated – in theory – for thresholds $\theta \in [-\infty, \infty]$. In practice, the ROC points are generated by scanning the classification output for all processed points when sorted in a decreasing threshold order, updating the number of TPs and FPs and hence TPR and FPR. This way of calculation - called *relative thresholding* - is made possible by exploiting the monotonicity of thresholded classifications [58], stating that instances classified positive by one threshold will stay positive for lower thresholds. The left subplot of Fig. 2.3 shows some important notions when analyzing an ROC curve. The green arrows shows the direction of threshold

decrease, with which TPs (FPs) increases from 0 to GT_+ (GT_-) going from bottom to top (left to right). The ideally performing classifier is represented by the point at the upper-left-hand corner where all positives are detected with zero false-alarms. On the other hand, the diagonal line ($TPR = FPR$) represents the worst-performing classifier acting equally to a randomly guessing algorithm (RGA). An RGA randomly classifies a proportion p of the testing set as positives leading to equal percentages of TP and FPs, *i.e.*, $TPR = FPR = p$. One should expect well performing algorithms to be represented by curves lying within the shaded area as near as possible to the upper-left-hand corner. An algorithm A is said to dominate another algorithm B if the curve representing B is totally contained in the area between the curve of A, the right and the bottom limits of the plot. For the dominance to be quantified, we need to interpolate between the calculated points of the ROC curve. This is equivalent to a convex hull drawn over the calculated points, generating a connected curve of the expected average performance lying between optimistic and pessimistic cases [58]. In that way, the true-positives and false-positives are assumed to grow linearly with respect to each other. To summarize the dominance in one value, researchers used the area under the ROC curve (ROC-AUC) which in contrast to the single-valued metrics, gives an idea of the algorithm's average performance independently from threshold choice. The area-under-the-curve is approximated by the sum of areas of successive trapezoids with each trapezoid covering the area between two successive points on the curve and the horizontal axis. The area of the trapezoid A_T from point A to point B is given by:

$$A_T = |FPR_B - FPR_A| \cdot \frac{(TPR_B + TPR_A)}{2} \tag{2.15}$$

The ROC-AUC can take values from 0 to 1 with the latter corresponding to the best performing algorithm. The worst performing classifiers are the ones lying around ROC-AUC = 0.5. Finally the curves lying outside the shaded area should be fixed by inverting the decision. Even if *relative thresholding* may be of use in theoretical comparison, in practice and for optimal threshold selection, absolute thresholding is more often adopted.

A significant limitation of ROC-based evaluation is encountered when applied to datasets with a large class imbalance. This happens because ROC and ROC-AUC are designed to be independent from the classes distribution in the testing datasets. Using TPR and FPR, they evaluate events and event-free instances separately and hence they are totally insensitive to class skewness [58]. This could be an asset for other problems. For instance, when having a large difference between the skew of the training and testing datasets or when evaluating the performance on datasets having different class skew. In case of NOD and HD, this is more of an inconvenience as in those applications, TNs do not add value. Added to that, the huge class imbalance inherent in both problems, *i.e.*,

between onsets and non-onsets, and howling and non-howling bins. For instance, in a howling ground-truth and especially when replacing the howling candidate selection with a full-spectrum DF calculation as explained in Chapter 8, there are many more negative than positive instances. The same is valid for NOD where music excerpts contains few samples marked as onsets compared to the total number of samples. A large change in the number of TP will result in a very small change in the FPR. This would usually result in ROC curves in which algorithms are represented by lines concentrated on the left side of the graph and are difficult to compare without zooming. Usually they are represented with an overly optimistic view when the skew is high [59]. Moreover, their relative ROC-AUCs will be totally misleading and not representative of the algorithm performance for problems where TNs are way less important than TPs [57].

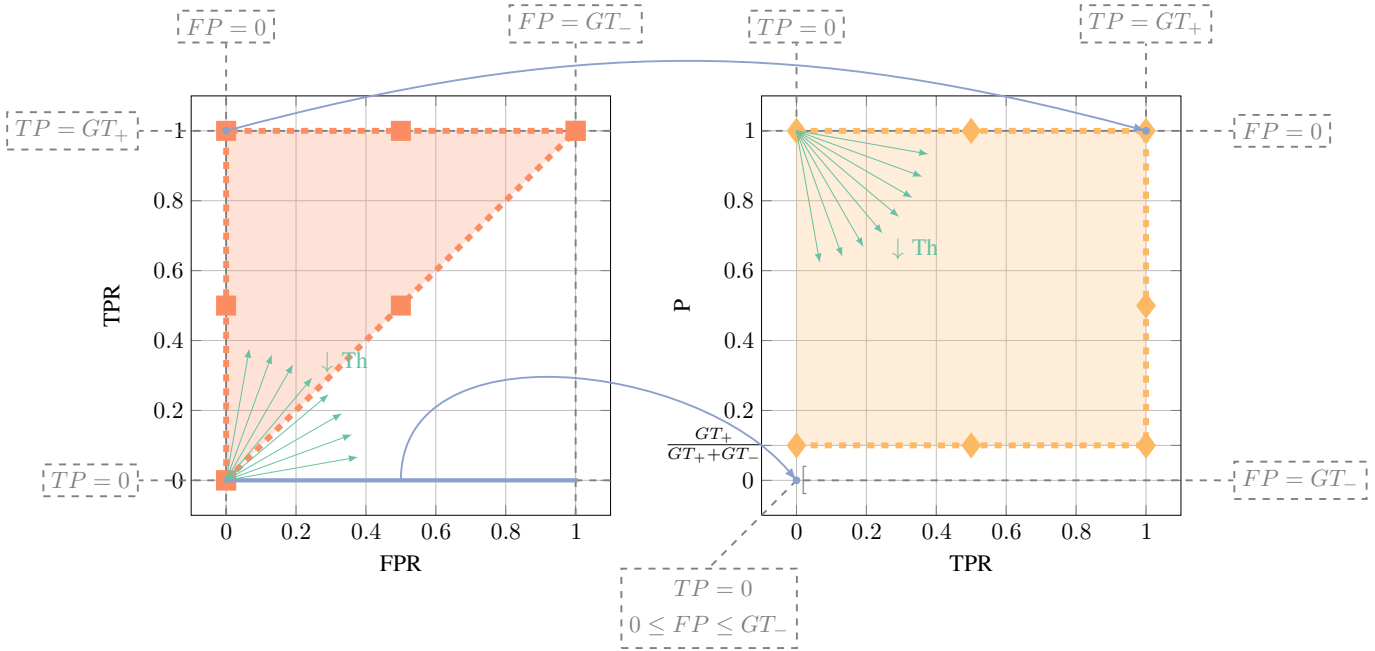


Figure 2.3: Moving from ROC to precision-recall (PR) graphs.

2.5.3 Precision-Recall (PR) curve: a better graph-based evaluation to handle skewed datasets

For the reasons stated previously, many researchers use a different graph-based method that is more sensitive to class imbalance – paying more attention to TPs, i.e. the quantity related to the target event – namely the PR curve [59]. For instance in the music information retrieval (MIR) field, PR analysis was used to evaluate the effect of the adaptive whitening on the NOD [60]. PR curve has the TPR on its horizontal axis under the name *recall*. On its vertical axis, it plots the *precision* which links the evaluation of the two classes instead of evaluating each class separately as with the TPR and FPR metrics. It is clear from Fig. 2.3, how it enhances the graph-based evaluation visibility as the – shaded – area relevant to the assessment is almost doubled in the PR curve compared to the ROC curve. This is valid for all problems with intrinsic high class imbalance as the lower limit of the shaded area is determined by the class skew. NOD and HD – as many other AED applications – are perfect examples of this kind of datasets. For instance, the classes' skew or event-to-non-event ratio of the different datasets used to evaluate the algorithms throughout the presented work amounts for 5:100 for NOD and 1:1000 for HD. Moreover, for quantifying the detection performance using a single valued metric, the area under the PR curve (PR-AUC) beats the ROC-AUC metric in comparing different features when the problem is highly imbalanced and the TNs are less important to the evaluation than TPs. Here the PR-AUC is giving the average precision while varying the recall. This renders the PR-AUC more correlated to F_1 than the ROC-AUC which is linearly correlated to Acc [57]. This approach expects a quasi-static training-to-testing classes distribution which is more or less the case for HD and NOD ground-truth data. The PR curve has a one-to-one correspondence to the ROC curve except when TP=0 and hence the *recall* R is equal to *zero*. This is shown in Fig. 2.3, where all points having TP=0 in the ROC curve collapse to one point $(0, 0)$ in the PR curve. Again by checking the threshold decrease directions and the TP and FP values, one should expect the well-performing algorithms to approach the top-left corner. Here the randomly guessing classifier, the baseline separating well-performing from poor-performing classifiers, is dependent only on the dataset's ground-truth classes distribution. Knowing that $TPR = FPR$ for a randomly guessing classifier, the PR baseline is the horizontal line $P = GT_+/(GT_+ + GT_-)$ derived

as follows:

$$P = \frac{TP}{D_+} = \frac{TP}{TP + FP} \cdot \frac{\frac{1}{GT_+ GT_-}}{\frac{1}{GT_+ GT_-}} = \frac{\frac{TPR}{GT_-}}{\frac{TPR}{GT_-} + \frac{FPR}{GT_+}} = \frac{GT_+}{GT_+ + GT_-}, \tag{2.16}$$

which is constant for all possible values of R. Another convenient property is that the dominance relation in ROC space is a consequence of the dominance in the PR space [59]. And again, for evaluating the dominance and the PR-AUC, we need to interpolate the intermediate points which is not a simple linear interpolation like with the ROC. However, the calculation is still based on the convex-hull concept considering the linear dependence between TP and FP applied locally for each two points. Practically, to interpolate between two calculated points A and B, the locally linear TP-FP slope

$$\alpha = \frac{FP_B - FP_A}{TP_B - TP_A} \tag{2.17}$$

is used to calculate an arbitrary intermediate point having the number of true positives equal to $TP_A + x$ as:

$$(R, P) = \left(\frac{TP_A + x}{GT_+}, \frac{TP_A + x}{TP_A + x + FP_A + \alpha x} \right). \tag{2.18}$$

At last, the PR-AUC is calculated similarly to ROC-AUC replacing (FPR, TPR) with (R, P).

2.5.4 Cross-validation scoring for small datasets

The majority of the available NOD and HD datasets are manually labeled. For a more robust result, the annotation is generally conducted by two or more experts and the result is averaged resulting in a longer labeling process. This is the main reason behind the small number of labeled samples in the available datasets. It is an important issue when evaluating the performance of the different algorithms, as the results will hardly generalize when using traditional evaluation which needs to divide the dataset into training, validation and testing. This evaluation method works best for large and representative datasets. Even though there are different initiatives to augment the datasets or to generate synthetic ones, it is still with small number of samples and the manual datasets are still the widely used ones.

On top of its use for parameters tuning, cross-validation is used to report performance when only small evaluation datasets are available. It works in the same way as for tuning. A K-fold cross-validation divides the datasets into K folds. The training is run K times, each time using $K-1$ folds, leaving out one fold for testing. At the end we get K testing results that are averaged and reported as the average score. This should be a better estimate for the algorithm performance on unseen data as it is less dependent on the dataset division into fixed training and testing sets.

2.6 Conclusion

This chapter started by formally defining AED and its various applications. It introduced the specific time-frequency structure of events studied in this thesis and how this links the two applications, NOD and HD. It has also introduced a detection methodology of three steps – initially proposed for NOD, together with an overview of the state-of-the-art solutions for each of its steps. The chapter ends by confronting different evaluation methods and metrics tackling the two problems: class-imbalance and limited availability of datasets which are two major issues for NOD and HD.

Part II

Onset Detection

Chapter 3

Automatic generation of synthetic music and annotations

The majority of note onset detection (NOD) results found in literature are obtained when testing with manually annotated datasets. The annotation process is generally conducted by at least two music experts. They are asked to listen to excerpts and inspect their corresponding waveforms and spectrograms in order to point out onset candidates by providing - per excerpt - a list of the onset candidate positions in time up to a certain time resolution. For this purpose, in [16] an annotation tool was developed allowing to listen to and analyze the time-frequency representation of a music excerpt simultaneously. The tool in [16] was used to annotate a small dataset which served as a basis for many researches during the past years, as well as for the Music Information Retrieval Evaluation eXchange (MIREX) challenge [61]. The final annotation is some kind of averaging of the experts' annotations. This may increase annotation robustness compared to a single annotator but still results in a subjective definition of ground-truth onsets. It depends not only on the experts' auditory and visual accuracy but also on the guidance given before and during the annotation experiment. Moreover, annotations will depend on the musical note context. For instance a lower intensity note following a louder one could easily be missed. Finally, the time needed for manual annotation is the main reason behind the limited number of available annotated datasets. For a fair and more accurate performance evaluation, we used instead a simple objective annotation method based on generating automatically annotated semi-synthetic

datasets. To this end, a tool called "*Mix-Notes*[1]" was developed – using MATLAB – to generate melodies out of isolated note recordings with their respective onsets (and possibly offsets) annotation. The main idea behind the tool and its different functionalities are summarized in Section 3.1, while Section 3.2 will provide more details on how the different NOD datasets used in this thesis were created.

3.1 Mix-Notes: a tool for NOD dataset generation

The development of Mix-Notes was instrumental to accommodate the different research questions studied in this thesis. The basic idea came from the note onset definition given by [16], as the first detectable part of the note in an isolated recording. Hence the tool performs automatic onset annotation in two steps: first automatically annotating the isolated note recordings, and then mix the note recordings as well as their annotations, thereby generating a musical excerpt with the corresponding automatically annotated onset locations. The idea behind Mix-Notes is similar to what is used by tools like SCAPER [62] or SimScene [63] to generate soundscapes, but focusing on doing music-scapes instead. Differently from SCAPER and SimScene, Mix-Notes needs first to detect and mark the onsets of the isolated notes before mixing them. Moreover, when dealing with music generation for training and evaluating NOD algorithms, a finer time resolution is needed than when dealing with general soundscapes.

At first, the tool loads isolated note – or chord – recordings for different instruments and playing styles. Then onset (and offset) ground-truth time locations are determined for every note based on a short-term energy measure. This is done easily and more accurately when notes are isolated, avoiding the note context dependency problem. Here, we used a simple energy measure: the onset is chosen to be the earliest point in time n_o at which the signal amplitude $|x(n_o)|$ satisfies that

$$|x(n_o)| \geq \frac{\rho}{100} \cdot \max_n \ |x(n)|, \quad 0 < \rho << 100, \tag{3.1}$$

where ρ is a percentage of the note's highest amplitude and should define the amplitude threshold of a just audible sound. After the annotation of the isolated note recordings, the tool generates a melody by linearly combining a set of chosen notes (*notes_list*) while imposing a time spacing between the onsets specified by the *onsets_spacing* input variable. Given this time spacing between onsets, the annotation for the created melody is easily generated from the isolated note recording annotations.

[1] code available at `https://gitlab.esat.kuleuven.be/Mina.Shehata/mix-notes`

As shown in Fig. 3.1, the current version of Mix-Notes is equipped with several parameters aiming to produce realistic and generic datasets. When the tool first opens, it prompts the user for an input folder which should be a library of isolated notes or sound events. For a smooth user experience, the different tool options are grouped into 4 panels, each focusing on a specific functionality and containing a short help file on the panel's main purpose. In the following, the panels are briefly explained pointing out the most relevant options and motivating their usage in generating the evaluation datasets.

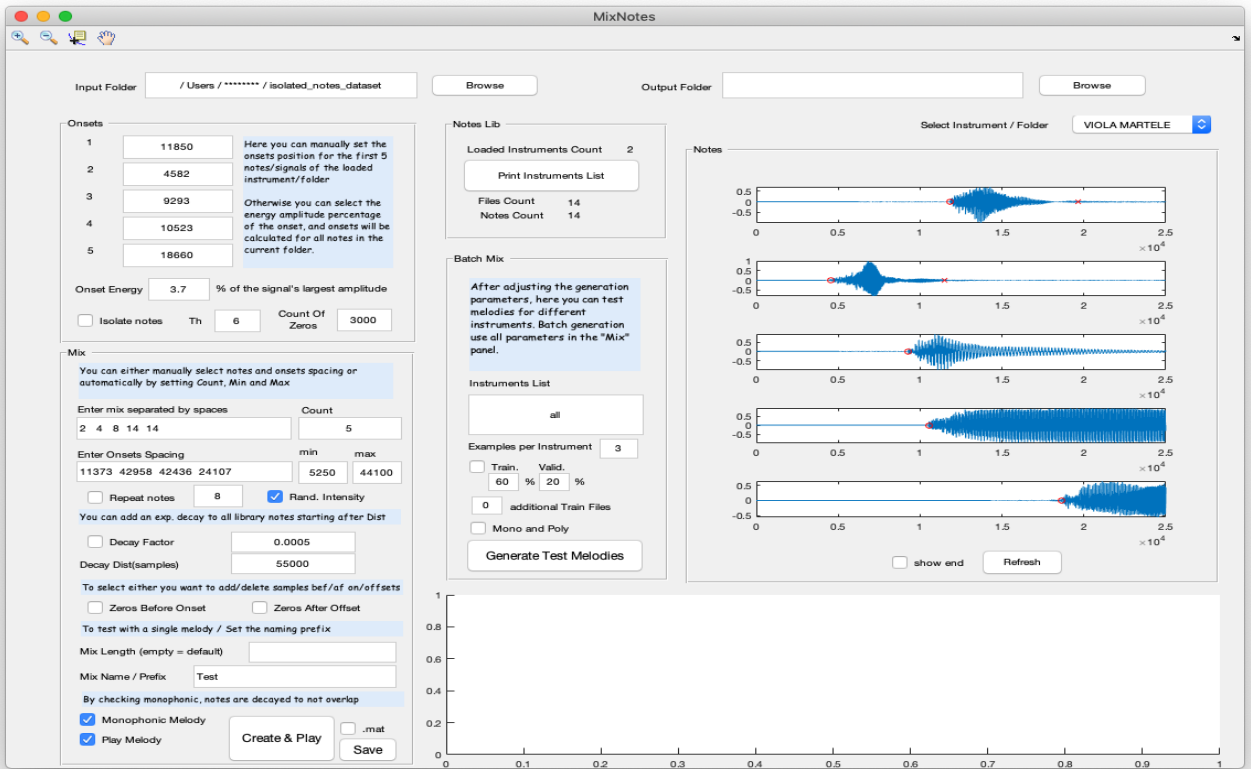


Figure 3.1: Screenshot of Mix-Notes GUI: a tool for mixing and automatically annotating music (onsets and offsets).

Notes panel: The tool treats each folder in the designated input folder as an instrument with a certain playing style. Using the *Instrument choice* drop-down menu, the user can choose to inspect a specific instrument. The *Notes* panel shown in Fig. 3.2 is plotting the waveform of 5 excerpts of the selected instrument. This panel serves as a visual aid when deciding on the value of ρ. It helps the user in finding an energy threshold ρ that accommodates the different instruments under test. This functionality should not be confused with manual annotations, as in this context it is for isolated notes and is mainly used to differentiate the real note onset from occasional recording noise.

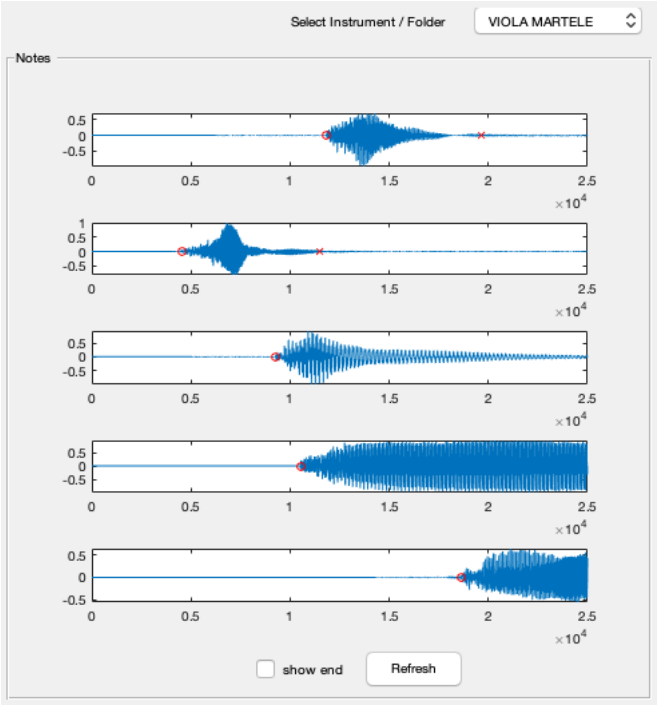


Figure 3.2: Notes panel.

Onsets panel: In Fig. 3.3, the onsets' sample number corresponding to the chosen ρ is shown for each of the 5 excerpts. The latest version of MixNotes includes an additional option to handle files containing more than one isolated note (sound) which is the case for some music or other type of events libraries. By ticking the *isolate notes* option and

setting its parameters, Mix-Notes will pre-process the files to isolate the events treating each as a separate note/event.

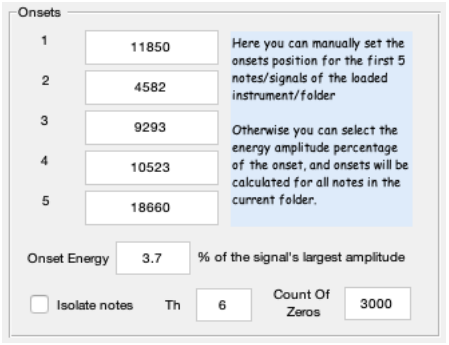


Figure 3.3: Onsets panel.

Notes Lib. panel (Fig. 3.4): This panel gives information about the loaded library differentiating between the number of files and the number of events.

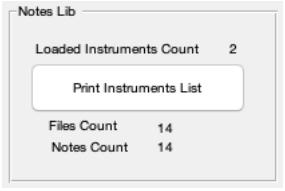


Figure 3.4: Notes Lib. panel.

Mix Notes panel (Fig. 3.3): This is practically the most important panel, responsible for mixing a new music excerpt out of isolated note recordings from a single instrument - *i.e.*, one folder in the library. For the scope of this work, only notes belonging to the same instrument folder were allowed to be mixed together. It starts by the *Notes list* which is an ordered list of C integer values representing the different notes to be mixed together. Then comes a $(C-1)$ ordered list holding the desired spacing in *samples* between each two successive notes in the mix. Users can choose to generate a random sequence of notes together with a random sequence of spacings between those notes by editing the notes *Count* variable. Several parameters are used when randomizing the sequence. The most relevant ones are listed below:

- *Min* and Max: These parameters define the minimum and maximum onset spacing values.
- *Repeated notes*: This option is selected in order to generate successive groups of repeated notes. In this case, the user can input how many times every note should be repeated. This kind of test examples is useful for challenging the state-of-the-art non-data-driven onset detection method *LogFiltSpecFlux* [45] which is using first-order differences. This detector would probably fail in detecting repeated notes with insufficient increase in magnitude at their onset.
- *Intensity randomization*: This is an option that is usually selected together with the previous one. In general the different isolated notes are normalized to a maximum amplitude equal to 1 and by selecting this option their intensities are randomly chosen between 50 % and 200 % of the intensity of the normalized signal.
- *Polyphonic vs. monophonic mixing*: By default the notes are summed respecting only the desired onset positions, thereby generating polyphonic mixes, *i.e.*, notes are overlapping in time. If monophonic is selected, *Mix Notes* forces an exponential decay (applying parameters provided by the user) on each note before the onset of the next note in order to simulate a monophonic mix. Care should be taken to not confuse this option with polyphonic instruments. Figures 3.6 and 3.7 illustrate these two ways of mixing for a violin excerpt where the onsets are marked by circles while offsets are marked by crosses.

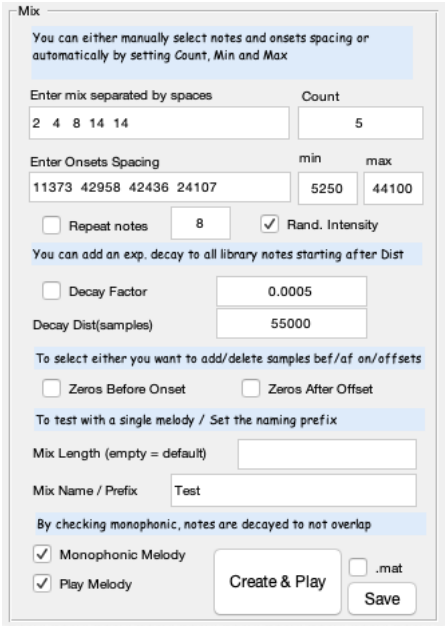


Figure 3.5: Mix Notes panel

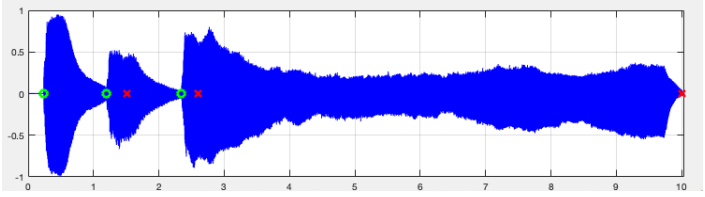


Figure 3.6: Monophonic mixing of a violin excerpt with 3 notes.

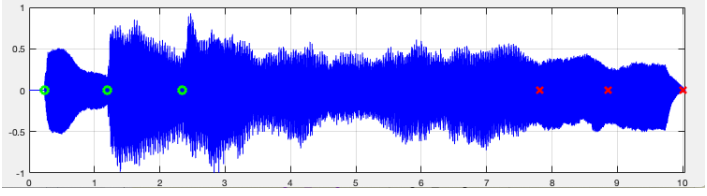


Figure 3.7: Polyphonic mixing of a violin excerpt with 3 notes.

Batch mix panel (Fig. 3.8): Finally this is the panel responsible for the evaluation dataset generation using the previously tuned randomization parameters. Users can specify the numbers of the instruments to use in a comma-separated list. Those numbers can be retrieved by pressing the *Print Instruments List* in the "Notes Lib" panel. Otherwise, all available instruments can be used for mixing by choosing the "all" keyword. If training and testing datasets are to be generated, the user can define the percentages of isolated note recordings used for each. In this way, note recordings used for training are separated from those used for testing. Finally, the user should specify the number of excerpts to be generated per instrument and may choose to have more training than testing samples.

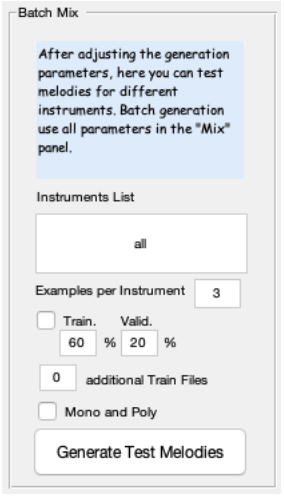


Figure 3.8: Batch mix panel.

Mix-Notes saves the output with the same folder structure as the input library. For each generated excerpt it saves the WAV file together with a comma-separated-values (csv) list of onset times in a ".txt" file sharing the same name of the ".wav" file. The tool provides a few more options for ground-truth fine-tuning, amplitude manipulations, playing and saving different extensions, that are easily understandable from the GUI.

3.2 Details of generated semi-synthetic datasets

To generate a semi-synthetic NOD dataset, we need a library of isolated musical note recordings. This kind of recordings are publicly available, but we specifically used the ones from the *McGill University Master Samples (MUMS)* library [64] which contains different instruments and playing styles. For this library, the energy percentage ρ value lying between 3 (for percussive instruments) and 7 (for long-attack instruments) is found to give realistic onset ground-truth values. Before using Mix-Notes to build our semi-synthetic NOD datasets out of the MUMS library, all files were checked for compatibility with the NOD framework. They were manually checked if they contain only one instance of the respective note otherwise they were neglected. For example instruments like the *harp* were not taken into consideration as recordings were mostly having many notes – *glissando* – in one file. All the percussions' rolling recordings and the folder "Percussions patterns and dance" were neglected for the same reason. Moreover, the percussions *skins* and *metals* folders were each divided into 3 folders as they contain much more notes than other instruments. This resulted in 138 folders containing different instruments with different playing styles. For an easier performance comparison, the folders were grouped in a similar grouping as the one used in the MIREX onset detection dataset. The grouping scheme is shown in Table 3.1 using similar naming as the MUMS folder names. The *bowed vibraphone* instrument is excluded from the grouping as it doesn't fit in any of the groups.

3.3 Conclusion

This chapter first reviewed the different limitations of the available datasets used for NOD algorithm training/tuning and evaluation. On the one hand, there is a limited amount of annotated examples as a result of the time-consuming and expensive annotation procedure. On the other hand, the labels are quite subjective as they are dependent on many factors: experts' auditory and visual accuracy, the annotation instructions and the musical note context.

As a solution, Chapter 3 introduces *Mix-Notes*[2], a MATLAB tool for generating semi-synthetic datasets with automatically annotated onsets. Inspired by the onset definition given in [16], the onsets are annotated for the notes when isolated before the mixing. Mix-Notes provides a variety of parameters to control different mixing options found interesting for the NOD task such as notes spacing, intensity, magnitude decay and repetitiveness. The tool can

[2]code available at `https://gitlab.esat.kuleuven.be/Mina.Shehata/mix-notes`

Table 3.1: Instruments Grouping

Pianos	
Harpsichord	8 Stop, 8_4 Stops, Buff Stop.
Piano	Concert Hall Steinway Soft, Hamburg Steinway Loud, Harmonics, Mpp Loud, Mpp Medium, Mpp Soft, Right Pedal Vol9, Steinway Plucked.
Bars & Bells	
Marimba	Grand Symphonic, Soft Mallet.
Steel Drum	Loud, Soft.
Vibraphone	Hard Mallet, Soft Mallet.
Others	Celesta, Crotales, Tubular Bells, Xylophone.
Brass	
French Horn	Normal, Muted.
Trombone	Pedal Notes, Tenor, Tenor Muted, Alto, Bass .
Trumpet	In C, In C_Harmon Mute, Bach, Bucket_Loud, Bucket_Soft, Cup_Loud, Hard Attack.
Others	Accordion Treble Notes, Cornet, Tuba.
Winds	
Sax	Alto , Alto Growls, Alto Screams, Baritone , Bass , Soprano , Tenor , Tenor Growls, Tenor Subtones.
Clarinet	Bflat, Bass, Contrabass, Eflat.
Flute	Alto_Non Vib, Alto_Vib, Bass_Flutter, Bass_Non Vib, Bass_Vib, Flutter, Non Vib, Vibrato.
Pan flute	High, Low, Medium.
Piccolo	Flutter, Non Vib, Vibrato.
Others	Bassoon, Contrabassoon, English Horn, Oboe.
Plucked Strings	
Acoustic Bass	Amplified, Pizz, Plucked, Slides.
Acoustic Guitar	Normal, Normal(2), Pizzicato, Sul Ponticello, Sul Tasto.
Electric Guitar	Normal, Harmonics, Stereo Chorus.
Others	Archluteoud, Cello Pizzicato, Renaissance 8 Course Lute, Viola Pizzicato, Violin Pizzicato, Vln Ensemble_Pizzicato-Wet.
Sustained Strings	
Acoustic Bass	Bowed_Vibrato, Harmonics, Martele, Muted.
Viol	Bass, Tenor, Treble.
Cello	Normal, Artificial Harmonics, Martele, Muted_Vibrato, Natural Harmonics, Non-Vibrato.
Viola	Artificial Harmonics, Martele, Muted, Natural Harmonics, Non-Vibrato, Vibrato.
Violin	1 Non Vibrato, 2 Non Vibrato, 3 Non Vibrato, Harmonics_Artificial, Harmonics_Natural, Martele, Muted_Vibrato, Vibrato.
Vln Ensemble	Dry-Bright, Harmonics-Wet, Martele-Wet, Soft Attack-Wet.
Percussions	
Acoustic Guitar Tapping, Metals-1/2/3, Skins-1/2/3, Tympani.	
Polypitched	
Electric Guitar	Dominant Ninths, Dominant Sevenths, Elevenths, Flat 7 Sharp 9, Major Seventh, Major Seventh Stopped, Major Sixths, Minor Sevenths, Ninths, Fifths.
Others	Accordion Chords.

be run in single-output mode for tuning or batch mode for generating large datasets. Chapter 3 ends by describing the basic dataset created for the different experiments of Part II with small variations for each, detailed in the respective chapter.

Further development of the tool may focus on intelligent music mixing. This can be done by applying music theory rules and enhancing the notes' fade-out. Moreover, more challenging excerpts can be generated by mixing different instruments together and having excerpts densely populated with onsets for the instruments which are played at faster tempo.

Chapter 4

Note onset detection based on a spectral sparsity measure

Musical note onset detection based on a spectral sparsity measure

Mina Mounir, Peter Karsmakers and Toon van Waterschoot

submitted for publication.

Abstract

If music is the language of the universe, musical note onsets may be the syllables for this language. Not only do note onsets define the temporal pattern of a musical piece, but their time-frequency characteristics also contain rich information about the identity of the musical instrument producing the notes. Note onset detection (NOD) is the basic component for many music information retrieval (MIR) tasks, and has attracted significant interest in audio signal processing (ASP) research. In this paper, we propose an NOD method based on the novel normalised identification of note onset based on spectral sparsity (NINOS^2) feature. The NINOS^2 feature can be thought of as a spectral sparsity measure, aiming to exploit the difference in spectral sparsity between the different parts of a musical note. This spectral structure is revealed when focusing on low-magnitude spectral components that are traditionally filtered out when computing note onset features. We present an extensive set of NOD simulation results covering a wide range of instruments, playing styles and mixing options. The proposed algorithm consistently outperforms the state-of-the-art Logarithmic Spectral Flux (LSF) feature for the most difficult group of instruments which are the sustained-strings instruments. It also shows better performance for challenging scenarios including polyphonic music and vibrato performances.

4.1 Introduction

Musical note onsets can be simply defined as the start of the notes and hence determine the temporal sctructure of music, but also play an important role in music color (timbre) perception [44]. This makes note onset detection (NOD) one of the most frequently encountered problems in the arising research fields of machine listening, music processing and music information retrieval (MIR). It can be formulated as an acoustic event detection (AED) problem with the signal under processing being a piece of music and the events being the note onsets.

Despite being a long-standing problem, literature does not offer a single but rather different definitions for note onsets. From a music performer's perspective, it is the time instant at which the peformer plays the note. Alternatively, when a note is decomposed into a transient (attack and decay) followed by a steady-state (sustain and release) component [65], an onset can be defined as the start of the note's attack [11]. Different definitions have led to different (often manual) labeling methods for onset ground truth generation, used in NOD algorithm design and evaluation. While the onset definition of [11] is the key behind the

proposed NOD feature, the definition found in [16] considering an onset as the first detectable part of a note in an isolated recording, will be considered in this paper for labeling the datasets used for tuning and testing NOD methods based on the proposed and state-of-the-art features.

Note onset detection is increasingly gaining research interest due to its usefulness in many music-related activities and applications: automatic music transcription [66], audio-to-score (audio-to-symbolic representation) alignment [67] , music analysis (tempo and beat tracking) [68] [69] and synthesis (enhancement of the attacks of synthesized notes) [70], instrument identification [44] and adaptive audio effects [71], e.g., applying different time stretching for transient and steady-state components of a note. Moreover, onsets are of great importance for music search engines and recommender systems when used as acoustic meta-data about songs [72].

A general scheme for NOD has been summarized in [11] where the main component is the onset detection function (ODF). The ODF describes the variation of an NOD feature over time, and typically represents a highly sub-sampled version of the original music signal from which onsets should be easily detectable as distinctive amplitude peaks [16]. In literature, two main groups of NOD methods can be distinguished: data-driven and non-data-driven methods. Data-driven methods build statistical models for note onsets by employing machine learning methods on a large set of training data. For instance, learning a probabilistic (hidden Markov) model exploiting the rhythmic regularity in music [73], training a neural network (recurrent [74], convolutional [34]), or learning dictionaries describing onset/non-onset patterns [67]. These machine learning methods can either solve a classification problem, differentiating onsets from non-onsets, or a regression problem of which the output is then used to estimate a suitable ODF. On the other hand, in non-data-driven methods the ODF is directly calculated from the analyzed signal or its extracted features. Even though data-driven methods have been shown to slightly outperform non-data-driven methods on the Music Information Retrieval Evaluation eXchange (MIREX) dataset [75], the former need to be trained on large annotated datasets in order to be generically applicable, which is currently impractical as most of the available datasets require manual annotation. Moreover non-data-driven methods are sometimes preferred over data-driven methods as the former allow more easily to find direct relations with the signal's musical properties compared to the latter.

Non-data-driven NOD methods [65] may operate in the time domain or frequency domain [45] and differ in the type of features extracted from the analyzed signal: magnitude, phase, energy, power spectral density, or time derivatives of these [44]. Many non-data-driven methods exist, see [11] for an early overview: envelope follower, high frequency content (HFC), spectral

difference, phase deviation, etc. For a more recent overview we refer to MIREX [75], where different methods are submitted and evaluated on the same dataset. According to the MIREX results [75], the state-of-the-art non-data-driven method employs the Complex Flux (CF) feature [56] which is based on the Super Flux (SF) feature [32] taking also phase information into consideration. Both features were proposed to tackle a specific but widely occurring problem (i.e., robustness to vibrato and tremolo effects) and share the basic properties of the Logarithmic Spectral Flux (LSF) feature [45], which in turn is based on the spectral dissimilarity or Spectral Flux (SpecF) feature proposed earlier in [44] but includes additional pre- and post-processing. This family of features results in an ODF that tracks the temporal evolution of the short-time spectrum based on first-order differences between spectral features in successive signal frames. Several enhancements were proposed to this family, for example adaptive whitening [60] to counter the effect of spectral roll-off. Another related NOD method is the constrained linear reconstruction method [76] which instead of calculating the first-order spectral differences, aims to minimize the spectral difference between the current frame and a group of previous frames and uses the residual as an ODF. These variations add some computational complexity with a slight performance improvement. As all these methods rely on differences between successive frames, the performance of these methods may drop when comparing successive notes sharing a considerable number of harmonics or with repeated notes having insufficient increase in magnitude at their onset. Finally, a more recent approach to NOD using pitch tracking was presented in [77] and performs better with pitched non-percussive instruments.

This paper provides a detailed treatment of a new non-data-driven NOD method based on the normalised identification of note onset based on spectral sparsity (NINOS^2) feature that was proposed earlier by the authors in the context of guitar note onset detection [33]. The NINOS^2 feature represents a (normalized) sparsity measure of the analyzed signal magnitude spectrum. Even though spectral sparsity implicitly contributes to operation of the previously discussed non-data-driven methods by differentiating between notes' transient and steady-state components, it has not been explicitly investigated. In [33] we focused on guitar notes and chord progressions. Here instead we show and analyze results covering a wider range of instruments and playing styles. Moreover, we provide an in-depth analysis of the NINOS^2 feature in terms of its sparsity properties, propose two new and improved variants of the feature, compare its computational complexity to state-of-the-art non-data-driven features, and show its superior performance in challenging NOD scenarios involving polyphonic music, repeated notes, and vibrato effects. Finally, this paper also provides a more detailed explanation of the novel annotation approach introduced in [33], in which semi-synthetic music excerpts are generated and automatically annotated out of isolated musical note recordings.

Having introduced the problem, the related work and challenges, Section 4.2 will discuss in further detail the related non-probabilistic methods and their general solution steps. Section 4.3 provides an in-depth treatment of the NINOS^2 feature and the resulting novel NOD method. The experimental evaluation is shown in Section 4.4 comparing NOD results with the NINOS^2 and LSF features. Finally, Section 4.5 presents the conclusion and hints for future work.

4.2 Related work

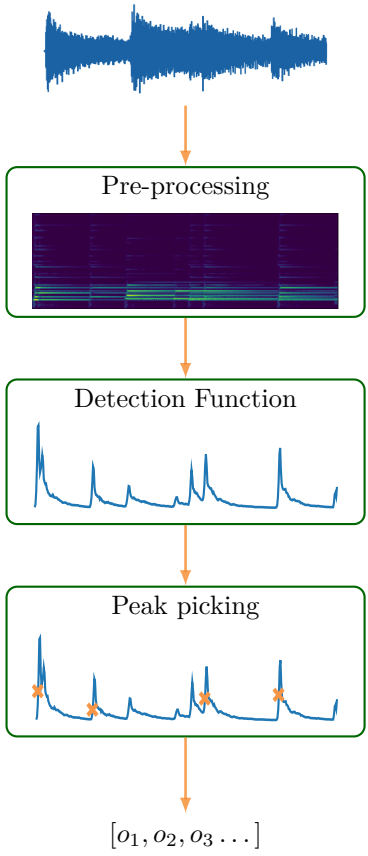


Figure 4.1: Solution scheme for NOD. An example of the output of the different steps is shown when NOD is applied on an input music excerpt.

The majority of non-data-driven NOD methods follow a general scheme that could be divided into three steps, see Fig. 4.1: pre-processing, reduction, and peak picking [11]. In the pre-processing step, the signal is processed in order to emphasize features related to onsets or to remove noise, which makes the detection easier. The second and most important step is the reduction step in which the ODF, sometimes alternatively named *novelty function*, is computed based on extracted signal features. The resulting ODF is then run through the peak-picking step, in which distinct points marking the estimated onset times are selected from the ODF.

As stated previously, the state-of-the-art non-data-driven methods are operating on spectral features of the input signal in order to detect spectral dissimilarities at note onsets. Their respective ODFs are calculated as the first-order difference of the spectral magnitude over successive frames, with some enhancements like adding phase information or adaptive whitening. The SpecF ODF is defined as follows[1],

$$SpecF(n) \;=\; \sum_{k=1}^{\frac{N}{2}-1} \left\{ H\left(|X_k\left(n\right)| - |X_k\left(n-1\right)| \right) \right\}, \tag{4.1}$$

where H is the half-wave rectification operator keeping track only of positive changes and neglecting the negative ones representing offsets,

$$H\left(x\right) \;=\; \frac{\left(x+|x|\right)}{2}, \tag{4.2}$$

$X_k(n)$ is the N-point short-time Fourier transform (STFT) at frame index n and frequency k of the windowed time-domain input signal $x\left(n\right)$,

$$X_k(n) \;=\; \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} w(m)x(n+m)e^{-\frac{2j\pi mk}{N}}, \tag{4.3}$$

and $w(m)$ is a length-N window function. In order to make the ODF invariant to signal scaling, in [37] the first-order difference was applied to the log-magnitude spectrum which is equivalent to the logarithm of the magnitude spectrum ratio between successive frames. In other words, instead of using the STFT magnitude coefficients $|X_k\left(n\right)|$ in (4.1), the STFT log-magnitude coefficients were used, defined as,

$$Y_k\left(n\right) = \log(\lambda|X_k\left(n\right)|+1)\;, \tag{4.4}$$

[1] In some publications, the SpecF ODF is defined by squaring the output of the half-wave rectification operator in (4.1) before summing over all frequency bins. Also, the lower summation index in (4.1) and (4.5) is sometimes chosen as $k = -N/2$ which results in a scaling of the resulting ODF with a factor of 2, due to the symmetry of the magnitude spectrum and the absence of a DC component.

where λ is a compression parameter and a constant value of 1 is added to ensure a positive value for the logarithm. The resulting LSF ODF is defined as

$$LSF(n) \;=\; \sum_{k=1}^{\frac{N}{2}-1} \left\{ H\left(Y_k\left(n\right) - Y_k\left(n-1\right)\right)\right\}, \tag{4.5}$$

This ODF generally gives better results compared to the SpecF ODF. We use the LSF ODF for comparison with the proposed method as it is generally considered the state-of-the-art non-data-driven method [45]. Optionally a filterbank according to the Western music scale can be applied as a pre-processing step before the LSF ODF calculation [45].

4.3 Proposed note onset detection method

4.3.1 Idea: transient vs. steady-state components

A musical note can be considered as a signal consisting of a transient component followed by a steady-state component, where each component can be expressed as a sum of sinusoids. From a sinusoidal modeling perspective, the key difference between the transient and steady-state components of a note, is the number of sinusoids needed to represent each. In fact, a transient requires a much larger number of sinusoids than a steady-state component to be accurately approximated. For some percussive instruments, the attack is nearly an impulse signal which conceptually corresponds to the sum of an infinite number of sinusoids. On the other hand, for non-percussive instruments, the transient is stretched over a longer time interval in which it typically also exhibits a broadband behavior which again leads to a representation requiring many sinusoids. This observation also follows from the definition of transients as short time intervals in which the statistical and energy properties of the signal change rapidly [65]. Consequently, the magnitude spectrogram of a musical note shows transients (attacks) that are spectrally less sparse than the subsequent steady-state (tonal) part. Spectral sparsity is considered here along the vertical (frequency) dimension of a magnitude spectrogram and indicates if few or many sinusoids are needed to represent the signal in that particular time frame.

In the following three subsections we will explain how this idea of spectral sparsity can be realized in the frame of the three-step non-data-driven NOD scheme introduced earlier. At this point, let us start by defining the input to the first step of this scheme. First, the analyzed music signal is divided into overlapping windowed frames and the STFT is computed as in (4.3). The

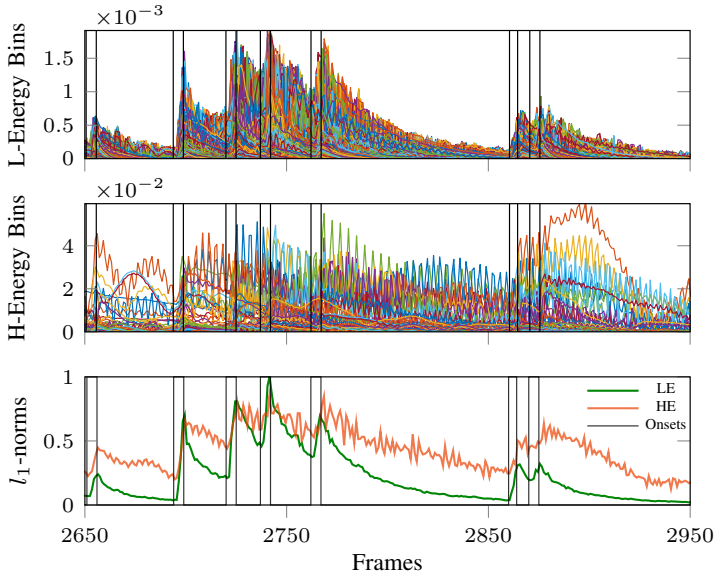
STFT log-magnitude coefficients $Y_k(n)$ are then calculated using (4.4), *i.e.*, in the same manner as for the LSF ODF.

4.3.2 Pre-processing: Coefficients subset

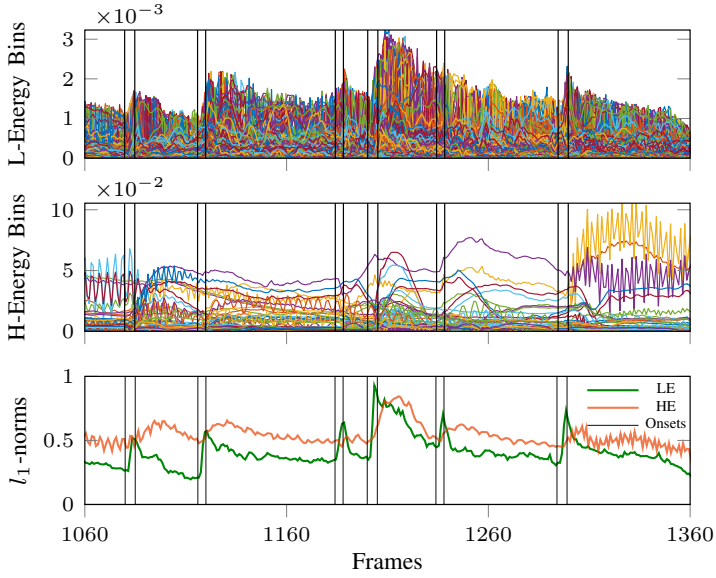
Before calculating the NINOS^2 ODF, we apply a pre-processing step to maximize the differentiation between the STFT log-magnitude spectra of frames containing onsets and other frames in terms of spectral sparsity. Instead of measuring the spectral sparsity of the entire STFT log-magnitude spectrum of each frame, only a subset of the STFT log-magnitude coefficients will be used. This is motivated as follows.

For the majority of pitched musical instruments, when looking at the magnitude spectrogram of an isolated note recording, it can be observed that the frequency components corresponding to the note's fundamental frequency and harmonics exhibit high energy during transients followed by a slow energy decrease after the transient. On the other hand, the remaining frequency components, *i.e.*, those not related to the note's fundamental frequency or harmonics, show a remarkable increase in energy during transients followed by a relatively fast energy decay, *i.e.*, faster than the post-transient energy decay of the harmonics.

To illustrate this observation, Fig. 4.2 compares the time variation over 300 signal frames of the energy per frequency bin, after splitting the frequency bins for each time frame into two subsets: one subset containing the frequency bins with low energy (L-Energy Bins, upper subplots in Fig. 4.2) and another subset containing the frequency bins with high energy (H-Energy Bins, middle subplots in Fig. 4.2). This is shown in Fig. 4.2 for three different instruments: electric guitar (Fig. 4.2a), cello (Fig. 4.2b) and trumpet (Fig. 4.2c). Despite the overlaying of many different curves in these plots, it can be clearly observed that the energy variation in the L-Energy Bins exhibits more pronounced peaks in frames containing onsets (marked by the vertical black lines in Fig. 4.2), compared to the energy variation in H-Energy Bins. While the energy in H-Energy Bins does not always increase when reaching an onset, the energy in L-Energy Bins does. Apart from this, it can also be observed that the shape of the energy decay curve after an onset seems to be characteristic for the type of instrument. For instance, the L-Energy Bin energy curves for the electric guitar and the cello do not show an explicit sustain component for their notes while the same curves for the trumpet do show a clear distinction between the transient (attack and decay) and steady-state (sustain and release) components. We can also notice, again from the L-Energy Bin energy curves, that the electric guitar has a faster attack and decay differentiating it from the other two instruments shown here. Finally, the lower subplots for each instrument in Fig. 4.2 show the

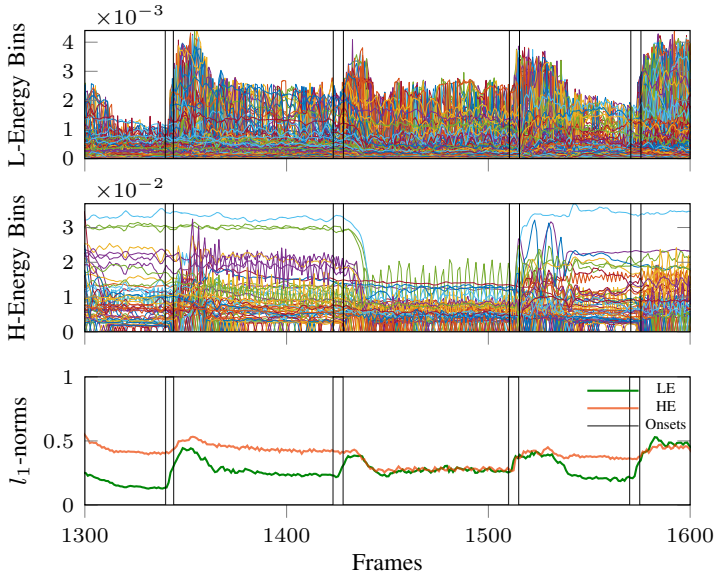


(a) **Electric Guitar** (major seventh stopped)



(b) **Cello** (non-vibrato)

Figure 4.2: Temporal variation of signal energy per frequency bin and of spectral sparsity for three excerpts from different instruments. Note onsets are indicated by vertical black lines. (Top & Middle) *Low-* and *High-energy* STFT log-magnitude coefficients variation. (Bottom) *Low-* and *High-energy* STFT log-magnitude coefficient vector ℓ_1-norm variation.



(c) **Trumpet** (Bach)

Figure 4.2: Temporal variation of signal energy per frequency bin and of spectral sparsity for three excerpts from different instruments. Note onsets are indicated by vertical black lines. (Top & Middle) *Low-* and *High-energy* STFT log-magnitude coefficients variation. (Bottom) *Low-* and *High-energy* STFT log-magnitude coefficient vector ℓ_1-norm variation (cont.).

time variation of the ℓ_1-norm of the vector of STFT log-magnitude coefficients, again separated into the subsets of H-Energy (HE) Bins and L-Energy (LE) Bins. Indeed, the ℓ_1-norm is one of the simplest sparsity measures [78] and illustrates in these examples that spectral sparsity may be used to detect onsets when applied to the subset of L-Energy Bins.

In summary, removing high-energy STFT log-magnitude coefficients before calculating a spectral-sparsity-based ODF, thus neglecting coefficients corresponding to fundamental frequencies and harmonics, will enhance the discriminative power of the ODF. It is important to emphasize how fundamentally different this pre-processing step is from existing NOD methods. Indeed, pre-processing often emphasizes signal components in H-Energy Bins, considering components in L-Energy Bins to be less relevant for NOD.

The frequency bin subset selection is implemented as follows: the $N/2-1$ STFT log-magnitude coefficients $Y_k(n)$, $k = 1, \ldots, N/2-1$ for frame n are sorted in ascending order and only the first J out of $N/2-1$ coefficients are used

afterwards, with

$$J = \left\lfloor \frac{\gamma}{100} \left(\frac{N}{2} - 1 \right) \right\rfloor, \tag{4.6}$$

where $\lfloor \cdot \rfloor$ denotes the floor function and γ represents the percentage of (one-sided) STFT frequency bins to be contained in the L-Energy Bins subset. These J coefficients are collected in a length-J vector $\mathbf{y}(n)$ and their frequency bin indices are collected in the set $\mathcal{I}_{\mathrm{LE}}(n)$, *i.e.*,

$$\mathbf{y}(n) = \left[Y_k(n) \right], \; k \in \mathcal{I}_{\mathrm{LE}}(n). \tag{4.7}$$

4.3.3 ODF: spectral sparsity feature

We now provide a detailed derivation of the NINOS^2 feature and its resulting ODF, starting from the idea of spectral sparsity introduced in Section 4.3.1. First, after having collected the STFT log-magnitude coefficients of the Low-Energy Bins subset during pre-processing, a spectral sparsity feature denoted in [33] as identifying note onsets based on spectral sparsity (INOS^2) is introduced. The INOS^2 feature is actually an inverse-sparsity measure as to yield large values for non-sparse frames, thus highlighting possible onset locations in time. Considering the INOS^2 feature per frame n, the INOS^2 ODF is obtained,

$$\Upsilon_{\ell_2 \ell_4}(n) = \frac{\|\mathbf{y}(n)\|_2^2}{\|\mathbf{y}(n)\|_4} = \frac{\sum\limits_{k \in \mathcal{I}_{\mathrm{LE}}(n)} Y_k^2(n)}{\left(\sum\limits_{k \in \mathcal{I}_{\mathrm{LE}}(n)} Y_k^4(n) \right)^{\frac{1}{4}}}. \tag{4.8}$$

We will first argue why the INOS^2 feature can indeed by interpreted as an inverse sparsity measure, before considering how it could be further improved for the purpose of NOD. While there is no universal consensus for defining and measuring sparsity, two basic conditions that should be satisfied by any sparsity measure have been stated in [78]: the most sparse signal is the one having all its energy concentrated in one sample while the least sparse is the one having its energy equally distributed over all its samples. For example, according to this definition, a measure S that satisfies the following inequalities for vectors of equal length would indeed be considered a sparsity measure,

$$\begin{aligned} S_{max} &= S([0,0,0,0,1]) > \dots \\ &> S([0,0,1,1,1]) > \dots \\ &> S([1,1,1,1,1]) = S_{min}. \end{aligned}$$

where S is a sparsity measure defined for vectors of equal length. Using this dummy example it can be verified that the INOS^2 feature in (4.8) is indeed an inverse sparsity measure. However, this observation only holds if the vectors in comparison have similar energies. For instance, according to the INOS^2 feature, the vector $v = [0, 0, 100, 100, 100]$ is less sparse than the vector $[1, 1, 1, 1, 1]$, which clearly contradicts with the basic conditions for a sparsity measure given above. The explanation for this behavior is that the INOS^2 feature is in fact a joint energy and inverse sparsity measure. This becomes clear when rewriting (4.8) as follows,

$$\Upsilon_{\ell_2\ell_4}(n) = \|\mathbf{y}(n)\|_2 \cdot \frac{\|\mathbf{y}(n)\|_2}{\|\mathbf{y}(n)\|_4}. \tag{4.9}$$

The first factor $\|\mathbf{y}(n)\|_2$ is the vector ℓ_2-norm which directly relates to the signal frame's energy. As onsets are usually accompanied by an increase in energy, see *e.g.*, Fig. 4.2, this property of the INOS^2 feature is actually desirable for NOD. In fact, the use of a pure energy measure was proposed as one of the earliest features for NOD, resulting in an ODF known as the envelope follower [44]. The second factor in (4.9), *i.e.*, the ratio between the signal frame's ℓ_2-norm and ℓ_4-norm, can be understood to measure inverse sparsity by applying the unit-ball concept [65] as shown in Fig. 4.3.

The figure shows a 2-D coordinate system where each point is representing a vector $\mathbf{v} \in \mathbb{R}^2$. Focusing on the first quadrant, it can be noticed that points near the X or Y axis, represent sparse vectors and inversely, while moving away from the axes towards the 45° line (*e.g.*, moving away from the X axis in the direction of the (solid) red arrow), vectors get less sparse. By applying the basic conditions for a sparsity measure given above, the most sparse vectors correspond to points lying on the axes, *e.g.*, $[0, 1]$ and $[1, 0]$, while points lying on the $\pm 45°$ lines correspond to the least sparse vectors, e.g. $[\sqrt{0.5}, \sqrt{0.5}]$ lying on the unit circle. Now let us consider the relation between $\ell_p - norms$ for different p in this 2-D coordinate system. The unit circle represents the ℓ_2-norm unit ball, *i.e.*, the set of points representing vectors having an ℓ_2-norm equal to 1. For each of the points on the ℓ_2-norm unit ball, we can graphically evaluate the ℓ_4-norm by looking at the ℓ_4-norm unit ball and its scaled versions each containing one of the points. We observe that the scaled ℓ_4-norm ball gets smaller, shrinking along the dashed red arrow, while considering a point moving along the (solid) red arrow. This means the ratio of the ℓ_2-norm and the ℓ_4-norm increases as the vector becomes less sparse.

Apart from the two basic conditions discussed above, a number of additional desirable properties of a sparsity measure have been proposed in [78]. These properties and their interpretation for a sparsity-based NOD feature are briefly summarized here:

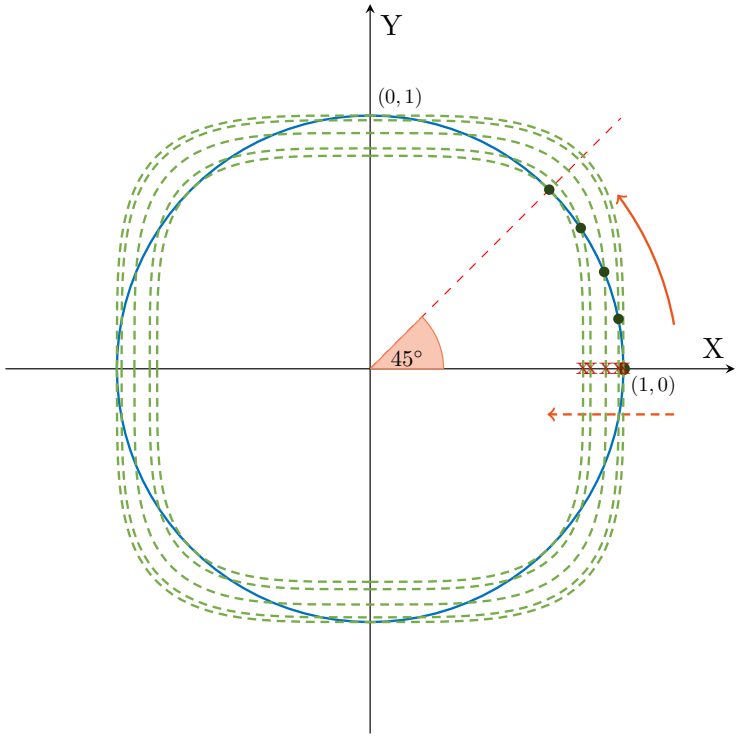


Figure 4.3: 2-D unit-ball illustration of the relation between the ℓ_2-norm and ℓ_4-norm.

- *Robin Hood*: Taking from the richest (highest-energy signal samples) and giving to the poor (lowest-energy signal samples) decreases the sparsity of a signal. Considering the analogy with spectral sparsity in the NOD context, the transient component of a note represents a scenario with many poor where energy is fairly distributed, after which the energy shifts to the harmonics in the steady-state component which represents a scenario with few rich.

- *Scaling*: Sparsity is independent of signal scaling, *e.g.*, $S([0, 0, 1, 1, 1]) = S([0, 0, 100, 100, 100])$. This is an important property in terms of decoupling an NOD feature into an energy measure and sparsity measure, as discussed above when factorizing the INOS^2 feature as in (4.9).

- *Rising Tide*: Adding a constant value to each of the signal samples decreases sparsity.

- *Cloning*: Sparsity measures should preferably assign the same value to cloned signals i.e. $S([0,1]) = S([0,0,1,1]) = S([0,0,0,1,1,1] = \dots)$. As will be discussed below, this property is relevant to reduce the amount of computations needed for the NOD feature calculation and to make the resulting ODF insensitive to the use of frames with different frame lengths.

- *Bill Gates*: As an individual gets richer, sparsity increases. In the context of NOD, as the steady-state component of a note shows a more pronounced tonal behavior, *i.e.*, the more its energy is concentrated in a few frequency components, it will be easier to discriminate the transient from the steady-state component using a spectral sparsity feature.

- *Babies*: Appending zeros to a signal increases its sparsity.

The ratio of a signal vector's ℓ_4-norm and ℓ_2-norm, *i.e.*, the inverse of the second factor of the INOS^2 feature in (4.9), can be understood to satisfy all of the above properties except for the cloning property. In other words, this ratio is sensitive to changes in the vector length, *i.e.*, J in the context of the INOS^2 feature. This is an undesirable property for an NOD feature for several reasons. Firstly, it makes the choice of a detection threshold and peak-picking parameters dependent on the frame size N and pre-processing parameter γ. Secondly, it makes the feature incompatible with a processing strategy in which successive signal frames may have different lengths to achieve detections with different time resolutions (which is however outside the scope of this paper). Finally, it would yield different feature values for the cases that the full frequency grid or only the positive frequency grid is used in the feature calculation, as briefly touched upon earlier[1].

For these reasons, the INOS^2 feature can be improved by normalizing the inverse-sparsity factor in (4.9) such that it yields a value $\in [0,1]$ independently of the length J of the vector $\mathbf{y}(n)$. Let us denote the inverse-sparsity factor in (4.9) as $\mathcal{S}(n) = \|\mathbf{y}(n)\|_2/\|\mathbf{y}(n)\|_4$ and its (theoretically achievable) minimum and maximum value as $\mathcal{S}_{\min}$ and $\mathcal{S}_{\max}$, respectively. A normalized version of $\mathcal{S}(n)$, *i.e.*, $\bar{\mathcal{S}}(n) \in [0,1]$ can then be obtained as follows,

$$\bar{\mathcal{S}}(n) = \frac{\mathcal{S}(n) - \mathcal{S}_{\min}}{\mathcal{S}_{\max} - \mathcal{S}_{\min}}. \tag{4.10}$$

The values $\mathcal{S}_{\min}$ and $\mathcal{S}_{\max}$ can be found by considering two extreme cases of an arbitrarily scaled length-J vector: the sparsest possible vector $[a,0,0,\dots,0]$ having $\mathcal{S}(n) = 1 \triangleq \mathcal{S}_{\min}$ and the least sparse vector $[a,a,a,\dots,a]$ having

$\mathcal{S}(n) = \sqrt[4]{J} \triangleq \mathcal{S}_{\max}$. By substituting these values in (4.10) we obtain that

$$\bar{\mathcal{S}}(n) = \frac{\mathcal{S}(n) - 1}{\sqrt[4]{J} - 1} = \frac{\frac{\|\mathbf{y}(n)\|_2}{\|\mathbf{y}(n)\|_4} - 1}{\sqrt[4]{J} - 1}. \tag{4.11}$$

Combining this normalized inverse-sparsity factor with the energy factor in (4.9) finally results in the normalized version of the INOS^2 feature denoted as NINOS^2. Considering the NINOS^2 feature per frame n, the NINOS^2 ODF is obtained,

$$\aleph_{\ell_2\ell_4}(n) \ = \frac{\|\mathbf{y}(n)\|_2}{\sqrt[4]{J} - 1} \left(\frac{\|\mathbf{y}(n)\|_2}{\|\mathbf{y}(n)\|_4} - 1 \right). \tag{4.12}$$

This definition of the NINOS^2 ODF slightly differs from the original definition in [33] because of the different normalization procedure. The two definitions deviate in particular for relatively small values of J, and only the definition proposed here in (4.12) guarantees that the inverse-sparsity factor $\bar{\mathcal{S}}(n) \in [0, 1]$.

Finally, in this paper we also propose a new sparsity-based NOD feature, starting from the observation that the above unit-ball demonstration can be applied to any ratio of norms, $\|\mathbf{y}(n)\|_p / \|\mathbf{y}(n)\|_q$, where $p < q$. A particularly interesting choice is $p = 1, q = 2$, since in this case the energy factor $\|\mathbf{y}(n)\|_2$ cancels out with the denominator of the inverse sparsity factor $\|\mathbf{y}(n)\|_1 / \|\mathbf{y}(n)\|_2$, and the joint energy and inverse sparsity measure reduces to the ℓ_1-norm, *i.e.*, (compare to 4.9),

$$\Upsilon_{\ell_1}(n) = \|\mathbf{y}(n)\|_2 \cdot \frac{\|\mathbf{y}(n)\|_1}{\|\mathbf{y}(n)\|_2} = \|\mathbf{y}(n)\|_1. \tag{4.13}$$

When applying the same normalization strategy as above, the ℓ_1-norm version of the NINOS^2 ODF is readily obtained,

$$\aleph_{\ell_1}(n) \ = \frac{\|\mathbf{y}(n)\|_2}{\sqrt{J} - 1} \left(\frac{\|\mathbf{y}(n)\|_1}{\|\mathbf{y}(n)\|_2} - 1 \right). \tag{4.14}$$

However, by comparing the expressions in (4.13) and (4.14), it is clear that the non-normalized expression is much cheaper to compute due to the cancellation of the ℓ_2-norm in (4.13). Therefore, we will further consider the INOS^2 (ℓ_1) ODF in (4.13) as a computationally appealing alternative to the NINOS^2 ($\ell_2\ell_4$) ODF in (4.12).

4.3.4 Peak-Picking

To allow for a fair comparison of the proposed ODF with the state of the art, the same peak-picking procedure as used with the state-of-the-art non-data-driven NOD method in [56] is adopted here. According to this procedure (shown

here for the NINOS^2 ($\ell_2\ell_4$) ODF and similar for the other ODFs), an onset is detected in the nth signal frame if *all* of the following conditions are satisfied,

1. $\aleph(n) = \max_l \ \aleph(n+l), \quad \text{with} \quad l = -\alpha, \ldots, +\beta,$
2. $\aleph(n) \geq \ \frac{1}{a+b+1} \sum_{l=-a}^{+b} \aleph(n+l) + \delta,$
3. $n - p > \Theta,$

where α, β, a, b and Θ are the peak-picking parameters defined using the following terminology, *before maximum*, *after maximum*, *before average*, *after average* and *combination width*, all counted in frame units, and p is the frame index of the previously detected onset. The interpretation of the above conditions is that (1) an onset should correspond to the highest ODF amplitude in the neighborhood of the frame index n, (2) an onset's ODF amplitude should be an amplitude offset δ above its neighborhood average ODF amplitude, and (3) an onset should be Θ frames apart from the closest preceding onset.

While the peak-picking parameter values are kept the same as in [56], the combination width Θ is set equal to the detection window length which is the maximum number of frames in which a single ground-truth onset could occur. This value depends on the frame overlap and is calculated using the following relations:

$$h = \lfloor (1-q)N \rceil, \tag{4.15}$$

$$r = f_s/h, \tag{4.16}$$

$$\Theta = \lceil rN/f_s \rceil, \tag{4.17}$$

where $\lfloor \cdot \rceil$ and $\lceil \cdot \rceil$ denote the nearest integer and ceiling functions, h is the frame hop size in samples, $q \in [0, 1]$ is the frame overlap factor, N is the frame size in samples, r is the frame rate, f_s is the sampling frequency and Θ as defined above is the number of frames to be skipped after one onset detection before aiming to detect a new onset. It is advisable to choose the value of Θ greater than the value given in (4.17) in case of instruments with very long attacks.

4.3.5 Onset Detection Parameters

For a complete understanding of the new non-data-driven NOD method based on the (N)INOS^2 feature and of the performance measures explained further, the most important parameters impacting the method's NOD performance are summarized here:

- *Processing frame size* (N): It should be larger than a single period of the signal [44] yet small enough to capture transients.
- *Detection resolution*: It depends on the frame rate r which is inversely proportional to the hop size h.
- *Processing mode*: The detection algorithm could be run in either *offline* or *online* mode. In the latter, the peak-picking parameters β and b are set to zero.
- *Evaluation window*: It is used to increase the detection window – implementing an onset ground-truth – to handle the lack of precision inherent in the onset annotation process, *i.e.*, onsets may occur slightly before or after the annotated ground-truth onset.

4.4 Experiment setup and evaluation

4.4.1 Dataset

As mentioned earlier, the majority of NOD results found in literature are obtained by testing on manually annotated datasets. A manual annotation process consists in asking music experts to listen to a number of music excerpts and to inspect their waveforms and spectrograms in order to find the onset times. The final annotation is obtained by averaging the different experts' onset times. Despite this averaging step, manual annotation still yields a subjective definition of ground-truth note onsets, as it depends not only on the experts' auditory and visual accuracy but also on the instructions and guidance provided when running the experiment. Moreover, these annotations will depend on the musical note context, for example, a lower intensity note following a louder one could easily be missed. Finally, manual annotation is time-consuming and costly, which is the main reason behind the limited number of available annotated datasets.

In order to achieve a fair, more objective and more accurate performance evaluation, we instead use automatically-annotated semi-synthetic datasets. To this end, a Matlab tool called "Mix Notes" has been developed by the authors to generate music excerpts from isolated note recordings together with their respective ground-thruth onset annotation. The tool loads isolated note and/or chord recordings from a database and then automatically determines the ground-thruth note onsets (and offsets) based on a short-term energy measure. This can indeed be done easily and accurately when notes are individually recorded and avoids the dependency of the annotation on the musical note context. Here, we used a simple energy measure: the onset is chosen to be the earliest point

in time n_o at which the absolute value of the signal amplitude $|x(n_o)|$ satisfies that

$$x(n_o) \geq \frac{\rho}{100} \cdot \max_n |x(n)|, \quad 0 < \rho << 100, \tag{4.18}$$

where ρ is a percentage of the note's highest amplitude and should define the amplitude threshold of a just audible sound. The tool then generates a melody by automatically mixing the selected note recordings while imposing a specified minimum time spacing between successive onsets. An annotation file for the generated melody is automatically created using the isolated note annotations with the timing information from the mixing process.

For the evaluation of the NOD methods in this paper we have created an automatically annotated semi-synthetic dataset using the MixNotes tool with isolated note recordings of various instruments and playing styles, starting from the recordings in the McGill University Master Samples (MUMS) library [64]. Before annotating the isolated note recordings and generating the semi-synthetic music excerpts, all files in the MUMS library were checked for compatibility with the NOD problem. The files were manually checked and retained only when containing just one instance of the respective note. For example, instruments like the *harp* were excluded as its recordings were mostly composed of many notes, *e.g.*, a glissando recording, in one file. All the percussions' rolling recordings and the library folder "Percussions patterns and dance" were excluded for similar reasons. Moreover, the *skins* and *metals* percussions folders were each divided into 3 folders as these contain many more note recordings than other instruments. This resulted in 138 folders containing different instruments with different playing styles. For a more structured performance evaluation, the folders were grouped using a similar grouping as used in the MIREX NOD dataset [75]. The grouping scheme is shown in Table 4.1 using similar naming as the MUMS folder names. The *bowed vibraphone* instrument is excluded from the grouping as it doesn't fit in any of the groups.

The proposed and state-of-the-art non-data-driven NOD methods are evaluated on these 138 folders covering a large number of instruments (electric, acoustic, wind, brass, strings,percussions, etc.) and playing styles (notes, chords, pizzicato, sul ponticello, bowed, screams, flutter, etc.). For each instrument and playing style, *i.e.*, each separate folder, three test melodies are generated, each consisting of a sequence of 50 randomly selected notes. The first melody is used for tuning the NOD algorithm parameters whereas the other two are used for performance evaluation. The time spacing between two notes in each generated melody is randomly chosen from a uniform distribution on the interval $[5250, 44100]$ samples which, at a sampling frequency $f_s = 44.1$ kHz, corresponds to $[0.12, 1]$ s. These values were chosen to limit the onsets per minute (opm) to 500, which is slightly above the maximum opm of an average pop song with a tempo of 120 beats per minute (bpm) and 4 notes per beat, *i.e.*, 16th notes.

Table 4.1: Instruments Grouping

Pianos	
Harpsichord	8 Stop, 8_4 Stops, Buff Stop.
Piano	Concert Hall Steinway Soft, Hamburg Steinway Loud, Harmonics, Mpp Loud, Mpp Medium, Mpp Soft, Right Pedal Vol9, Steinway Plucked.
Bars & Bells	
Marimba	Grand Symphonic, Soft Mallet.
Steel Drum	Loud, Soft.
Vibraphone	Hard Mallet, Soft Mallet.
Others	Celesta, Crotales, Tubular Bells, Xylophone.
Brass	
French Horn	Normal, Muted.
Trombone	Pedal Notes, Tenor, Tenor Muted, Alto, Bass .
Trumpet	In C, In C_Harmon Mute, Bach, Bucket_Loud, Bucket_Soft, Cup_Loud, Hard Attack.
Others	Accordion Treble Notes, Cornet, Tuba.
Winds	
Sax	Alto , Alto Growls, Alto Screams, Baritone , Bass , Soprano , Tenor , Tenor Growls, Tenor Subtones.
Clarinet	Bflat, Bass, Contrabass, Eflat.
Flute	Alto_Non Vib, Alto_Vib, Bass_Flutter, Bass_Non Vib, Bass_Vib, Flutter, Non Vib, Vibrato.
Pan flute	High, Low, Medium.
Piccolo	Flutter, Non Vib, Vibrato.
Others	Bassoon, Contrabassoon, English Horn, Oboe.
Plucked Strings	
Acoustic Bass	Amplified, Pizz, Plucked, Slides.
Acoustic Guitar	Normal, Normal(2), Pizzicato, Sul Ponticello, Sul Tasto.
Electric Guitar	Normal, Harmonics, Stereo Chorus.
Others	Archluteoud, Cello Pizzicato, Renaissance 8 Course Lute, Viola Pizzicato, Violin Pizzicato, Vln Ensemble_Pizzicato-Wet.
Sustained Strings	
Acoustic Bass	Bowed_Vibrato, Harmonics, Martele, Muted.
Viol	Bass, Tenor, Treble.
Cello	Normal, Artificial Harmonics, Martele, Muted_Vibrato, Natural Harmonics, Non-Vibrato.
Viola	Artificial Harmonics, Martele, Muted, Natural Harmonics, Non-Vibrato, Vibrato.
Violin	1 Non Vibrato, 2 Non Vibrato, 3 Non Vibrato, Harmonics_Artificial, Harmonics_Natural, Martele, Muted_Vibrato, Vibrato.
Vln Ensemble	Dry-Bright, Harmonics-Wet, Martele-Wet, Soft Attack-Wet.
Percussions	
Acoustic Guitar Tapping, Metals-1/2/3, Skins-1/2/3, Tympani.	
Polypitched	
Electric Guitar	Dominant Ninths, Dominant Sevenths, Elevenths, Flat 7 Sharp 9, Major Seventh, Major Seventh Stopped, Major Sixths, Minor Sevenths, Ninths, Fifths.
Others	Accordion Chords.

The above dataset generation procedure is repeated four times to produce four datasets with distinct properties relevant to NOD performance evaluation:

- *M* dataset: monophonic mixing.
- *MR8* dataset: monophonic mixing with forced repeated notes where every note is repeated 8 times.
- *P* dataset: polyphonic mixing.
- *PR8* dataset: polyphonic mixing with forced repeated notes where every note is repeated 8 times.

The *M* and *P* datasets (and equivalently the *MR8* and *PR8* datasets) are composed of the same note sequences with the same onset times and time spacings. However, in the *M* and *MR8* datasets, an exponential amplitude decay is applied to the sustain part of the notes, such that each note has faded out before the next note starts, effectively forcing a monophonic music excerpt. In the *P* and *PR8* datasets, successive notes are allowed to overlap in time. The use of repeated notes in *MR8* and *PR8* is mainly to assess the algorithms' performance in detecting onsets of successive notes that are sharing a considerable number of harmonics with insufficient increase in magnitude at their onset. These latter datasets are expected to be challenging for the state-of-the-art algorithm based on spectral differences.

4.4.2 Performance measures

An important issue when assessing detection performance is to decide how true positives and negatives are counted. Here we adopt the same approach as used in the evaluation of the state-of-the-art method [32], where two onsets detected within one detection window are counted as one true and one false positive, and a detected onset can count only for one detection window.

The most common way to compare NOD methods is by evaluating their relative F1-scores. The F1-score is defined as the harmonic mean of the precision (P) and the recall (R) given by:

$$F1 = \frac{2PR}{P+R}, \tag{4.19}$$

with the precision being the ratio of correctly detected onsets ("true positives") to the total number of onsets under test, and the recall comparing the number of correctly detected onsets to the total number of points detected as onsets.

Since a true positive could occur anywhere within the detection window, a new measure is developed to determine how large the detection window should be in order to achieve the reported F1-score. For each true positive in the music excerpt under test, the time difference of the ground-truth onset frame to the first frame that is covered by the detection window is saved, and then the *Detections standard deviation* σ_d, *i.e.*, the standard deviation of the time difference of the ground-truth onset to the start of the detection window, is computed for each music excerpt used in the evaluation. When σ_d is small, the algorithm will be detecting onsets in the same relative position to the start of the detection window. This is an important measure that reflects how well the algorithm detects the different onset times relative to each other.

4.4.3 Computational complexity and memory usage

The running time of a non-data-driven NOD method is comparable when using the LSF or NINOS^2 ODFs, and considerably shorter when using the INOS^2 (ℓ_1) ODF. Whereas the computation of the NINOS^2 feature is slightly more expensive than the LSF feature, mainly due to the required pre-processing, the opposite is true for the memory usage as the NINOS^2 feature does not require any information from the previous frame. Here we compare the type and number of operations and the amount of memory needed for the computation of the three features, omitting the calculation of the STFT log-magnitude coefficients and the peak picking which are shared by all methods. Note that for all features we only use half minus one, *i.e.*, $\frac{N}{2}-1$, of the FFT coefficients.

The number of operations and memory positions needed for the LSF feature calculation is

$$\mathcal{C}_{\mathrm{LSF}} = \left(\frac{N}{2}-1\right) Sub + \left(\frac{N}{2}-2\right) Add + \left(\frac{N}{2}-1\right) Comp$$

$$\mathcal{M}_{\mathrm{LSF}} = N-2$$

with *Sub*, *Add*, and *Comp* representing scalar subtractions, additions, and comparisons respectively. Here, the comparisons are needed for the half-wave rectification. The number of operations and memory positions needed for the (N)INOS^2 feature calculation is (with $J < \frac{N}{2}-1$, see (4.6)),

$$\begin{aligned}\mathcal{C}_{\mathrm{NINOS}^2(\ell_2\ell_4)} = {} & (2J+1)Mult + 2(J-1)Add + 3\ Sqrt \\ & + 1\ Sub + 1\ Div + \left(\frac{N}{2}-1\right)\log\left(\frac{N}{2}-1\right) Comp\end{aligned}$$

$$\mathcal{C}_{\text{INOS}^2(\ell_1)} = (J-1)Add + \left(\frac{N}{2}-1\right)\log\left(\frac{N}{2}-1\right)Comp$$

$$\mathcal{M}_{\text{(N)INOS}^2} = \frac{N}{2}-1$$

with *Mult*, *Sqrt*, and *Div* representing scalar multiplications, square roots, and divisions, respectively. Here, the comparisons are needed for the frequency-bin sorting in the pre-processing. Note that the computation of the ℓ_1-norm in (4.13) does not require the calculation of absolute values since the elements of $\mathbf{y}(n)$ are positive by construction, see (4.4).

4.4.4 Parameter tuning

The non-data-driven NOD methods presented here, are dependent on a number of parameters: input preparation, pre-processing, reduction, and peak picking parameters. Finding the optimal combination of parameter values is not an easy task. In this paper, the different parameters are tuned in a sequential manner, in other words the parameters are selected one after another. We will now summarize how the different parameters have been tuned, making use of the tuning dataset introduced in Section 4.4.1.

The input test melodies are sampled with a sampling frequency $f_s = 44.1$ kHz. Each melody is divided into overlapping frames of $N = 2048$ samples (46.4 ms). This frame size value was chosen by looking at the resulting spectrograms for three different frame sizes: 1024, 2048 and 4096. These are all powers of 2 to benefit from using the Fast Fourier Transform (FFT). By repeating this for various instruments, $N = 2048$ was found the best trade-off between frequency and time resolution in (visually) emphasizing onsets in the spectrogram. By maximizing the F1-score over the tuning dataset, we found the Hanning window offering a better performance than the Hamming window, and the Discrete Fourier Transform (DFT) outperformed the Discrete Cosine Transform (DCT) in computing the STFT coefficients. The compression parameter λ was set to 1 as in the Madmom implementation [79]. A frame overlap of 90%, *i.e.*, $q = 0.9$, is used similarly to the state-of-the-art method, resulting in a frame rate of $r = 215$ frames-per-second (fps) or a 4.6 ms detection resolution which is better than the temporal hearing resolution (≈ 10 ms). The evaluation window is set to ± 25 ms around the ground-truth onset.

Due to the fact that the NINOS^2 feature satisfies the cloning property (D4 in [78]), using only 50 % of the STFT coefficients will give the same results. Note that this does not hold for the non-normalized INOS^2 feature. The percentage γ of STFT frequency bins used to compute the (N)INOS^2 feature is set to value

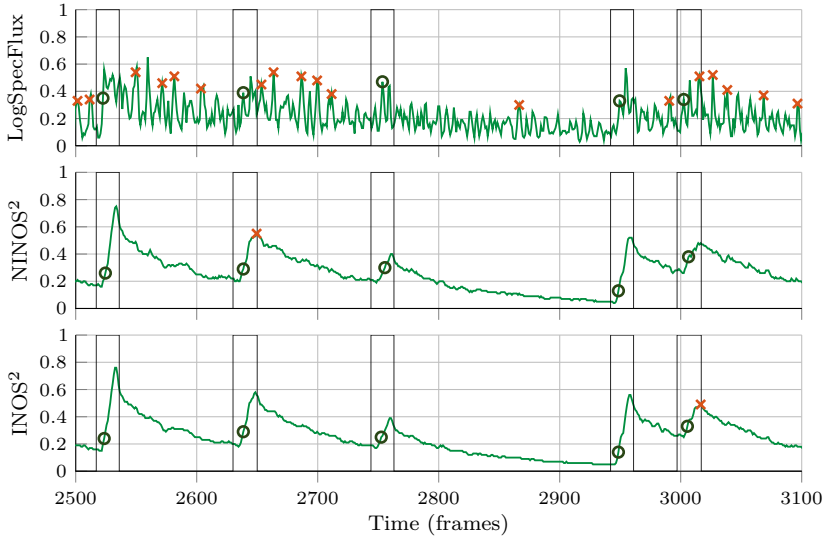
$\gamma = 95.5$ %, which was found to maximize the F1-score over the tuning dataset for γ values lying between 90 and 99.

The peak-picking parameters $\alpha = 30$ ms and $a = 100$ ms are kept as in [56] as these values resulted in better F1-scores for all the algorithms compared to the values found in [79]. The parameters β and b are set to zero to allow the NOD methods to operate online. The parameter Θ is set equal to the detection window length as stated earlier. Finally, the amplitude offset δ – sometimes referred to as the detection threshold – is tuned per feature to maximize their performance on the tuning dataset.

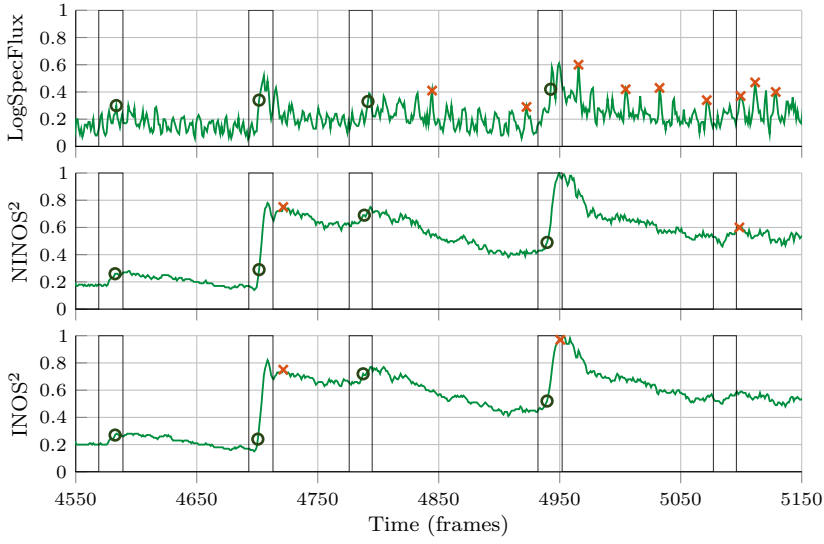
4.4.5 Results and discussion

In this section the performance of three non-data-driven NOD methods, using the proposed NINOS^2 ($\ell_2\ell_4$) ODF in (4.12), the proposed INOS^2 (ℓ_1) ODF in (4.13), and the state-of-the art LSF ODF in (4.5) is compared. The pre-processing introduced in Section 4.3.2 is only applied for the proposed methods NINOS^2 and INOS^2, but not to LSF. Before considering more quantitative results, we first look at a few examples of the different ODFs and the onset detections resulting from the peak picking. This is illustrated in Fig. 4.4 for three different instruments belonging to different instrument groups: the electric guitar (Fig. 4.4a), the cello (Fig. 4.4b) and the trumpet (Fig. 4.4c). The examples chosen here are excerpts taken from the P dataset.

Onset detection windows indicating ground-truth onsets are marked with vertical black lines. While circles are used to mark true positives in the peak-picking results, false positives are marked with crosses. Finally the false negatives are easily noted by unmarked detection windows. By visually analyzing these figures, it can be observed that the (N)INOS^2 ODF is remarkably smoother than the LSF ODF and presents higher amplitudes at onset times for the different instruments. When moving from the easier percussive instrument in Fig. (4.4a) to the more complicated non-percussive instruments in Figs. (4.4b) and (4.4c), it can be seen that the LSF ODF fails to show a clear amplitude rise at onset times. Both a high amplitude and a fast-rising amplitude at onset times are two beneficial ODF properties to have onsets successfully detected in the peak-picking step. As a consequence, the LSF ODF in Fig. 4.4 yields a higher number of false positives and false negatives after peak picking, compared to the (N)INOS^2 ODF. A final remark that clearly stands out in the guitar example, is that true positives are detected in the (N)INOS^2 ODF on the rising edge of an ODF amplitude peak rather than at the maximum of the amplitude peak. This is a consequence of the

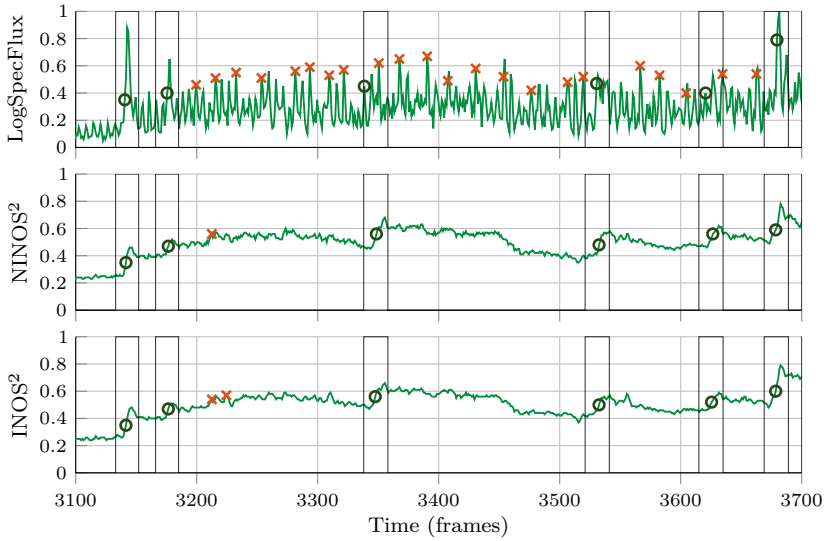


(a) **Electric Guitar**



(b) **Cello**

Figure 4.4: Comparison of ODFs and peak-picking results for three example music excerpts.



(c) **Trumpet**

Figure 4.4: Comparison of ODFs and peak-picking results for three example music excerpts (cont.).

smoothness of the ODF, resulting in an earlier onset detection which is a useful property for online NOD algorithms.

After this qualitative comparison, let us take a more quantitative approach by evaluating the previously described performance measures F1-score and detections standard deviation σ_d. The values of these measures are shown in Tables 4.2, 4.3, 4.4 and 4.5, summarizing the results obtained for the different datasets introduced earlier (*M*, *P*, *MR8*, *PR8*). In each table, while the upper rows show the results for the different instruments organized by instrument groups, the lower rows show the results for a subset of the instruments organized into two groups depending on whether they are played with or without vibrato. The instruments included in this subset are cello, viola, violin, flute, flute alto, flute bass and piccolo. Each table is vertically split into two parts. While the first part compares the F1-scores for the different methods, the second part compares the detections standard deviation σ_d. Finally, the *Total* row shows the weighted average of the results for all instrument groups where the weights are determined as the relative number of instruments per group.

Considering both the F1-score and the detections standard deviation σ_d results, a few trends can be observed. The state-of-the-art LSF ODF generally performs

Table 4.2: NOD performance measures for M dataset

	F1-score			σ_d		
	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF
Instruments Grouping						
Pianos	0.9115	0.9344	**0.9662**	1.6842	1.6525	1.5148
BarsAndBells	0.9871	**0.9923**	0.9766	1.2426	1.1936	0.8662
Brass	0.5331	0.5212	**0.6518**	2.8180	2.6832	2.7831
Winds	0.5053	0.4902	**0.6236**	3.4137	3.3547	3.1496
PluckedStrings	0.9362	0.9508	**0.9607**	1.5969	1.5633	1.3865
SustainedStrings	**0.5289**	0.5026	0.5083	2.4713	2.5047	2.8176
Percussions	0.8111	0.8303	**0.9351**	1.6992	1.5812	0.9619
Polypitched	0.6756	0.6487	**0.7419**	2.8399	2.7064	2.8293
Total	0.6700	0.6623	**0.7263**	2.4443	2.3937	2.3459
Vibrato Grouping						
Vibrato	0.3984	0.3851	**0.4065**	3.0771	2.9681	3.9253
Non vibrato	0.4821	0.4627	**0.6184**	3.4342	3.4319	2.9572

Table 4.3: NOD performance measures for P dataset

	F1-score			σ_d		
	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF
Instruments Grouping						
Pianos	0.8736	0.8981	**0.9111**	1.8126	1.7993	1.7189
BarsAndBells	0.9886	**0.9907**	0.9719	1.2774	1.2232	1.0028
Brass	**0.4365**	0.4360	0.3612	3.9106	3.9905	4.1966
Winds	0.4328	**0.4365**	0.3652	4.2315	4.1217	4.5686
PluckedStrings	0.9372	0.9482	**0.9491**	1.5887	1.5871	1.5686
SustainedStrings	0.4433	**0.4447**	0.3845	3.6015	3.6368	4.2535
Percussions	0.8021	0.8304	**0.9256**	1.8838	1.7340	1.0332
Polypitched	**0.7148**	0.6836	0.5113	2.8322	2.5303	3.6377
Total	0.6221	**0.6259**	0.5784	3.0426	2.9975	3.2867
Vibrato Grouping						
Vibrato	**0.3240**	0.3125	0.2670	4.9782	4.8557	5.2357
Non vibrato	**0.4394**	0.4229	0.3305	4.3878	4.2892	4.7833

Table 4.4: NOD performance measures for *MR8* dataset

	F1-score			σ_d		
	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF
Instruments Grouping						
Pianos	0.8968	0.9175	**0.9229**	1.7070	1.6610	1.4504
BarsAndBells	0.9943	**0.9953**	0.9765	1.2073	1.1742	0.9470
Brass	0.5459	0.5453	**0.6257**	2.8460	2.8731	3.4872
Winds	0.5323	0.5294	**0.5970**	3.4198	3.3137	3.4385
PluckedStrings	0.9520	**0.9655**	0.9628	1.5984	1.5710	1.4760
SustainedStrings	**0.5762**	0.5612	0.5344	2.6696	2.7099	3.0523
Percussions	0.8157	0.8379	**0.9256**	1.7448	1.6519	1.0948
Polypitched	0.6931	0.6770	**0.7042**	2.9939	2.9402	2.7570
Total	0.6915	0.6909	**0.7161**	2.5085	2.4775	2.5662
Vibrato Grouping						
Vibrato	**0.4555**	0.4533	0.4055	2.7296	2.6760	3.7082
Non vibrato	**0.4908**	0.4781	0.4849	3.7505	3.6386	4.1155

Table 4.5: NOD performance measures for *PR8* dataset

	F1-score			σ_d		
	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF	NINOS^2 ($\ell_2\ell_4$)	INOS^2 (ℓ_1)	LSF
Instruments Grouping						
Pianos	0.8864	**0.9127**	0.8831	1.8040	1.7966	1.7402
BarsAndBells	0.9889	**0.9937**	0.9743	1.2381	1.1880	0.9634
Brass	0.4798	**0.4825**	0.4455	4.0768	4.0931	4.5098
Winds	**0.4694**	0.4677	0.4415	4.3367	4.2574	4.4708
PluckedStrings	0.9503	**0.9617**	0.9518	1.5997	1.5798	1.5501
SustainedStrings	**0.5219**	0.5203	0.4660	3.8391	3.8170	4.4102
Percussions	0.8303	0.8534	**0.9184**	1.8487	1.7305	1.0665
Polypitched	**0.7318**	0.7167	0.6210	2.9600	2.8119	3.4602
Total	0.6593	**0.6630**	0.6312	3.1469	3.1003	3.3230
Vibrato Grouping						
Vibrato	0.3901	**0.4014**	0.3535	4.8887	4.8500	5.2842
Non vibrato	**0.4555**	0.4458	0.3628	4.1196	4.3297	4.7532

better on monophonic music excerpts, whereas the (N)INOS^2 ODF is more suited for polyphonic music excerpts. Performance varies strongly across instrument groups, and for the most challenging group of sustained-strings instruments, which exhibit non-percussive and slow-attack onsets, the (N)INOS^2 ODF outperforms the LSF ODF in all datasets. For datasets involving repeated notes, the performance gap between the different methods is slightly reduced, but the same observations can be made regarding monophonic vs. polyphonic performance for the F1-score. The detections standard deviation σ_d for repeated notes is however smaller with the (N)INOS^2 ODF for both monophonic and polyphonic excerpts. For the most challenging dataset in which polyphonic excerpts with repeated notes are considered (*PR8*), the (N)INOS^2 ODF outperforms the LSF ODF for all instrument groups except percussions. Moreover, the (N)INOS^2 ODF seems more robust to vibrato than the LSF ODF, which can be observed in particular in the monophonic datasets (*M* and *MR8*), where the performance improvement of the (N)INOS^2 ODF over the LSF ODF is clearly larger for the vibrato group than for the non-vibrato group. Finally, the performance of the NINOS^2 ($\ell_2\ell_4$) ODF and the INOS^2 (ℓ_1) ODF is fairly similar, with a consistent trend of the NINOS^2 ($\ell_2\ell_4$) ODF to perform slightly better for non-percussive instruments and the INOS^2 (ℓ_1) ODF for percussive instruments.

4.5 Conclusion and future work

In this paper we have proposed two new variants as well as a thorough analysis of the (N)INOS^2 spectral-sparsity-based NOD feature introduced earlier in [33]. Instead of focusing on the fundamental frequency and harmonic components of a musical note, as traditional non-data-driven NOD methods do, the (N)INOS^2 feature uses the subset of low-energy frequency components which are found to contain valuable information on note onsets. By investigating how measures for energy and sparsity can be combined and normalized, two new spectral-sparsity-based NOD features are introduced, as well as their related ODFs describing the feature variation over time. These can be combined with a state-of-the-art peak-picking procedure to obtain a novel non-data-driven NOD method.

Simulation results for a newly developed automatically annotated semi-synthetic dataset include a large and diverse set of instruments, playing styles and melody mixing options are reported in terms of the F1-score and a newly introduced measure, the detections standard deviation σ_d. In terms of the F1-score, the NOD performance using the LSF ODF is better for monophonic music excerpts, whereas the (N)INOS^2 ODF performs better for polyphonic music excerpts. In terms of the detections standard deviation σ_d, the benefit of the LSF ODF

over the (N)INOS^2 ODF for monophonic excerpts disappears for scenarios with repeated notes. Both performance measures illustrate that the (N)INOS^2 ODF is more suitable than the LSF ODF for the most challenging instruments group, *i.e.*, sustained-strings instruments, and playing style, *i.e.*, vibrato performance.

As the proposed INOS^2 (ℓ_1) ODF is considerably cheaper to implement than the proposed NINOS^2 ($\ell_2\ell_4$) ODF and the state-of-the-art LSF ODF, both in terms of computational complexity and memory usage, it seems to be the preferred feature to use for non-data-driven NOD.

Future work could include a performance comparison on publicly available manually annotated NOD datasets. Also, in future experiments, datasets should preferably be equilibrated in terms of instruments, *i.e.*, the different instrument groups should be populated similarly, which was not the case for the results presented in this paper. Finally, an instrument-dependent parameter tuning of the different methods and its impact on the resulting performance is worth investigating.

Chapter 5

CNN-based onset detection using synthetic data augmentation

CNN-based Note Onset Detection using Synthetic Data Augmentation

Mina Mounir, Peter Karsmakers and Toon van Waterschoot

Published in *in Proc. 28th European Signal Process. Conf. (EUSIPCO '20), Amsterdam, Netherlands, Aug. 2020.*

Abstract

Detecting the onset of notes in music excerpts is a fundamental problem in many music signal processing tasks, including analysis, synthesis, and information retrieval. When addressing the note onset detection (NOD) problem using a data-driven methodology, a major challenge is the availability and quality of labeled datasets used for both model training/tuning and evaluation. As most of the available datasets are manually annotated, the amount of annotated music excerpts is limited and the annotation strategy and quality varies across data sets. To counter both problems, in this paper we propose to use semi-synthetic datasets where the music excerpts are mixes of isolated note recordings. The advantage resides in the annotations being automatically generated while mixing the notes, as isolated note onsets are straightforward to detect using a simple energy measure. A semi-synthetic dataset is used in this work for augmenting a real piano dataset when training a Convolutional Neural Network (CNN) with three novel model training strategies. Training the CNN on a semi-synthetic dataset and retraining only the CNN classification layers on a real dataset results in higher average F_1-score (F_1) scores with lower variance.

5.1 Introduction

Many music processing and music information retrieval (MIR) applications rely on an elementary music analysis task known as note onset detection (NOD). In this task the detection algorithm is expected to produce an ordered list of time instants marking the start of the successive notes constituting a music excerpt. The majority of NOD solutions follows a three-step processing methodology: pre-processing, reduction function estimation, and peak-picking. Most of the recent and ongoing work focuses on the middle step, the objective of which is to yield an onset detection function (ODF): a highly sub-sampled version of the original signal, highlighting the note onsets while suppressing irrelevant signal details. Existing algorithms for ODF estimation fall into two main categories: data-driven and non-data-driven. The latter represents the traditional solutions exploiting the transient nature of a musical note onset in contrast to its subsequent steady-state portion or some other onset properties. These algorithms are completely independent from the data to be processed, as opposed to the data-driven algorithms. Checking the results reported in the Music Information Retrieval Evaluation eXchange (MIREX) NOD challenge [61], the Convolutional Neural Network (CNN) suggested by [34] is currently the best performing data-driven algorithm in terms of F_1-score (F_1).

A crucial problem in the design of data-driven NOD algorithms is the amount and quality of the available development datasets. For instance, the previously mentioned CNN model [34] is fitting around 290K parameters using one of the largest NOD datasets, made available by the authors upon request. This dataset was obtained by combining different datasets, resulting in a total size of 102 minutes (321 files, 28K onsets). A processing rate of 100 frames-per-second (fps) was used and the results reported in [34] were based on an 8-fold cross-validation, which implies the model was trained on 7/8 of the entire dataset for each of the folds. This translates to 89.25 minutes or 535K training points, which is hardly twice the number of model parameters. Despite the number of training points being far from the rule-of-thumb of 10 times the number of model parameters, a well performing model was obtained in [34], presumably thanks to regularization which is applied explicitly using dropout technique or sometimes implicit depending on the optimizer choice. Nevertheless, as for any data-driven model, it is always beneficial to have more training data, which is difficult to achieve for the NOD task because most of the available NOD datasets are manually labeled. Manual labeling is typically performed in two steps, listening to the music excerpt and then visually assessing its time-frequency representation, and is usually carried out independently by two or more expert annotators. For these reasons it is a costly and time-consuming operation. Moreover, as the resulting annotations are subjective and context-dependent, these should be used with care. This is usually handled by employing an evaluation window of about 50 ms around the onset ground truth [80].

One way to tackle the data availability problem, in particular for deep-learning-based NOD algorithms, is data augmentation. To our knowledge, the only work done for NOD dataset augmentation is reported in [81], where it was suggested to increase the model complexity by two modifications to the model proposed in [34]. These two modified models were trained by using data augmentation strategies common in music applications: transposition, time stretching, spectral envelope transposition, and remixing of sinusoidal and noise components. Unfortunately the modified models did not yield an enhanced NOD performance, and sometimes even resulted in a performance degradation.

The objective of the current paper is to propose a different strategy for tackling the data availability problem, based on the generation of automatically annotated semi-synthetic music excerpts. We assess the effect of using datasets comprising such excerpts on the performance of deep-learning-based NOD algorithms. More specifically, three model training strategies are proposed and evaluated. In the first strategy, the model is simultaneously trained on a semi-synthetic and a real dataset. In the second and third strategy, a Universal Onset model (UOM) is trained on a semi-synthetic dataset, which is then retrained (in two different ways resulting in two different strategies) on a real

dataset.

The paper is organized as follows. The data-driven approach to NOD is introduced in Section 5.2, focusing on CNN modeling and data annotation. The proposed model training strategies are described in Section 5.3. Section 5.4 introduces the different datasets used, and shows initial results regarding the use of single vs. multiple instrument types in the semi-synthetic dataset. Details of the CNN model [34] are provided in Section 5.5, after which the main results are presented in Section 5.6 and conclusions are formulated in Section 5.7.

5.2 Data-driven modeling and annotation

5.2.1 Data-driven NOD with CNN model

In the data-driven approach to NOD, a model is trained based on input data consisting of music excerpts each containing multiple note onsets, and a corresponding annotation consisting of the ground truth note onset times. In this paper, we will consider a CNN model (detailed in Section 5.5) for NOD which can indeed be considered the state-of-the-art data-driven solution [61]. A CNN is usually composed of interleaved convolutional and pooling layers, constituting the feature-learning layers, followed by a few fully connected layers forming the classification layers. Alternatively, these classification layers can be replaced by a Support Vector Machine (SVM) or a K-Nearest Neighbours (KNN) classifier.

Considering the fact that NOD is a binary classification problem with an extremely imbalanced distribution of classes (i.e., there are many more time instants without note onsets than with note onsets), the F_1 score is an appropriate evaluation metric. It is defined as the harmonic mean of the precision (P) and recall (R), i.e.,

$$F_1 = \frac{2PR}{P+R}. \tag{5.1}$$

A crucial question when assessing the CNN performance on small datasets is how well the reported scores will generalize to other evaluation datasets. One way to address this question is to report the scores after a K-fold cross-validation instead of using a straight train-validate-test workflow which would yield a good generalization only when working with large datasets. In a K-fold cross-validation, the dataset is first divided into K folds, after which K train-test rounds are performed. In each of the K rounds, it is common to use $K-1$ folds for training and 1 fold for testing. The final score is the average

of the K test scores. In this work, we adopt a different data split for the K-fold cross-validation, using 1 fold for training and $K-1$ folds for testing. Such data split has been shown useful to assess the performance of data-driven models in applications where a limited amount of datasets is available [82], as is typically the case in data-driven NOD. In other words, this data split is used to emulate the small amount of onsets-annotated excerpts and its limited variability compared to the music universe.

5.2.2 Manually vs. automatically annotated datasets

As mentioned above, the manual annotation of note onsets is a time-consuming and cumbersome procedure. Automatic annotation is therefore an attractive alternative, which could yield larger and more consistently annotated datasets. A first approach to automatic annotation consists in building datasets using musical instruments that have both an audio and MIDI output (e.g., the MAPS dataset [83]). This is however only possible for a limited range of instruments, i.e. (semi-)electronic instruments such as the Yamaha Disklavier used in [83].

A second approach, proposed in this paper, is to generate semi-synthetic datasets in which the music excerpts are composed by mixing recordings of isolated notes for which the onsets are automatically annotated *prior to mixing*. Adopting the definition of a note onset being "the first detectable part of the note in an isolated recording" [16], it can be understood that a simple energy measure suffices to automatically annotate note onsets in isolated note recordings. In this way, we can generate large datasets with consistent annotations obtained at minimal cost. Even if the proposed approach strongly depends on the availability of datasets with isolated note recordings, a large amount of semi-synthetic NOD datasets can be generated from just a limited amount of isolated note recordings, by varying the mixing parameters. Also note that we apply a random mixing strategy (i.e., random note selection, random ordering and spacing of notes), hence resulting in excerpts with somehow no musical structure. This can be justified by observing that data-driven NOD algorithms generally do not rely on such structure.

5.3 Proposed model training strategies

When using semi-synthetic datasets for NOD, one should be cautious not to overfit the model to the way the semi-synthetic data were generated. To this end, we propose three strategies to train a CNN model for NOD based on both real and semi-synthetic data. As a benchmark, we will compare the models

resulting from these three strategies with models trained on only real or only semi-synthetic data, resulting in the following five model training strategies, the performance of which will be compared in Section 5.6:

- **Real (R):** the model is trained only on real music excerpts. The excerpts could be manually annotated as in, e.g., [34] or alternatively, automatically annotated when using MIDI-supporting instruments as in, e.g., [84].
- **Synthetic (S):** the model is trained only on an automatically annotated semi-synthetic dataset.
- **Synthetic + Real (S+R):** the first proposed model training strategy consists in simultaneously training the model on excerpts from both datasets, by randomly interleaving real and semi-synthetic excerpts in the training set.
- **Synthetic pre-train followed by Real retrain ($S \triangleright R$):** in the second and third proposed model training strategy, the model is first pre-trained on a semi-synthetic dataset then retrained on a real dataset. This strategy resembles transfer learning, adapting a UOM trained on semi-synthetic data to real music excerpts. Given the CNN structure composed of one or more feature layers (convolution, pooling) followed by a number of classification layers (fully connected), see Section 5.2.1, the retraining in this strategy is run only on the classification layers as it can be assumed that the feature layers learned from one dataset can be used to generate useful features for another dataset as well. Moreover, this approach to retraining also protects the model from overfitting to a specific dataset. Two flavors of this strategy are proposed and evaluated in this paper for a CNN having two feature layers, see Section 5.5:
  - **Partial freeze ($S \triangleright R_1$):** in this strategy only the first feature layer is frozen, leaving the second layer open to learn additional features that couldn't be learned only from the semi-synthetic dataset. Hence we will use the symbol $S \triangleright R_1$ to denote this strategy of freezing the first feature layer only.
  - **Full freeze ($S \triangleright R_F$):** here, the model is only relearning the classification layers, assuming the semi-synthetic dataset suffices to learn a generic feature model for note onsets. This full freezing strategy is denoted by the symbol $S \triangleright R_F$.

When training a data-driven model on datasets from different origins, it is important to consider how to deal with their respective annotations. Specifically for NOD evaluation, we suggested in our recent work [85] to take into account

the time shift in annotations between the different datasets especially when training a Deep Neural Network (DNN). For instance, a semi-synthetic dataset may consistently have its onsets annotated earlier or later than a real dataset. Here, we adopt the approach of [85], by treating the *annotations time shift* as a hyperparameter determining the optimal temporal alignment of dataset annotations. Practically, given two datasets A and B from different origins, this hyperparameter is tuned by using the model learned on dataset A and estimate the annotations time shift δ that, when applied to the whole dataset B, maximizes the performance of this model on dataset B. When training a model on dataset B, the annotations B are then time-shifted by $-\delta$.

5.4 Datasets

Two semi-synthetic datasets were generated and tested in the frame of this paper. First, a large semi-synthetic dataset was generated for a wide range of instruments and it was termed All Instruments Semi-Synthetic (AISS) dataset. It was created using the isolated recordings for 138 different combinations of instrument and playing style from the McGill University Master Samples (MUMS) library [64]. The available notes per instrument and playing style were divided into 3 groups: 60% for training, 20% for validation and 20% for testing. Each of these groups were then used to produce a number of semi-synthetic music excerpts by mixing the isolated notes: 4 training mixes, 1 validation mix and 1 testing mix per instrument and playing style. Every mix contained 150 randomly selected notes with random inter-onset distances resulting in a tempo uniformly distributed in the range $[60, 2646]$ beats per minute (bpm). The note amplitudes were also randomized and the mixing was polyphonic, i.e., notes could (partially) overlap in time.

Despite the large amount of annotated examples, the optimization of the CNN model parameters using the AISS dataset stalled rapidly – after 10 epochs – for both the training and validation, see Fig. 5.1. In other words, the model was not able to continue learning which we believe may relate to the strong diversity of onset characteristics over the different instruments in the AISS dataset.

A second semi-synthetic dataset was therefore created, consisting of excerpts with only one type of instrument, namely the piano. This so-called Piano Semi-Synthetic (PSS) dataset was built from the piano recordings in the MUMS library, corresponding to 8 different pianos and playing styles or conditions: Concert Hall Steinway Soft, Hamburg Steinway Loud, Harmonics, Mpp Loud, Mpp Medium, Mpp Soft, Right Pedal Vol9 and Steinway Plucked. The PSS dataset generation process and parameters were exactly as for the AISS dataset,

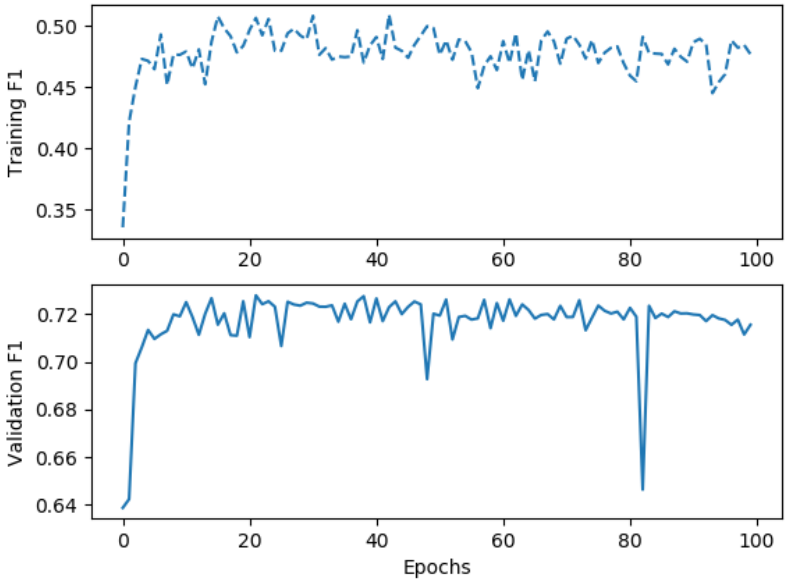


Figure 5.1: Comparison of training and validation F_1-scores for CNN model trained on the All Instruments Semi-Synthetic (AISS) dataset.

except for the amount of training mixes which was increased to 8 mixes to have a sufficiently large training set (see below). Figure 5.2 compares the validation and testing F_1 scores for the PSS and AISS datasets after training for 100 epochs, selecting the hyperparameters (annotations time shift and detection threshold) yielding the best validation F_1, and using the resulting model and hyperparameters to obtain the test F_1 scores. The box plots in the figure show the F_1 distribution over the different instruments/playing styles. It is clear that there is a higher average performance and less variance when restricting the NOD task to a single instrument type. Therefore, in the remainder of the paper we will focus on datasets of a single instrument type, and will use the PSS dataset as the semi-synthetic dataset when assessing the proposed model training strategies.

Further, two real datasets were selected from the MAPS dataset [83], more specifically the ENSTDkCl and ENSTDkAm datasets which will be referred to here as MAPS_CL and MAPS_AM. Also these two datasets are of the single instrument type (piano) and were recorded using a Yamaha Disklavier with

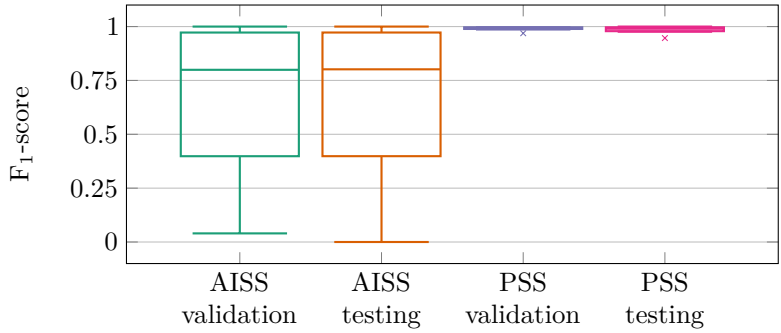


Figure 5.2: Comparison of validation and testing F_1-scores for CNN model trained on synthetic datasets (piano and all instruments).

Table 5.1: Dataset properties

Name	Files	Time (min.)	Points (K)	Onsets (K)	
				B.C.	A.C.
AISS	828	597.3	3583	124.2	122.4
PSS	88	125.4	752	13.2	13.1
MAPS_CL	30	136.1	753	76.4	41.8
MAPS_AM	30	125.5	817	78.0	37.2

annotations generated automatically from the output MIDI file. The suffixes "CL" and "AM" refer to *close* and *ambient* recording, hence specifying the distance between the microphone and the instrument. Table 5.1 provides a summary of the datasets introduced here, in terms of number of music excerpts (Files), total duration (Time), number of input points fed to the CNN with 100 fps processing, and amount of onsets per dataset. As in most other NOD research, onsets less than 30 ms apart are combined into a single onset for evaluation, therefore the last column in Table 5.1 differentiates between number of onsets before combination (B.C.) and after combination (A.C.).

Even though from Table 5.1 it may seem that the PSS dataset is smaller than the MAPS datasets, that is not the case when considering the data subsets

used for training. The PSS dataset is composed of 80% training, 10% validation and 10% testing excerpts. The training subset of length 113 min (676K points, 11.8K onsets) will be the set "S" used in the model training strategies proposed in Section 5.3. On the other hand, each of the MAPS datasets is divided for every training/validation/test round in the K-fold cross-validation (explained in Section 5.2), into 20% training, 10% validation and 70% testing. Hence, the MAPS training subset length amounts to about 25–27 min, which is 4 times less than the PSS training subset length.

5.5 CNN Model

The experiment will be run using the CNN model proposed in [34], which scored the highest F_1-score on the MIREX onsets dataset [61]. Briefly, a CNN is a feed-forward neural network with special hidden layers named convolutional layers. They add to a basic Neural Network (NN) the capability to retain spatial layout of the input which translates to time and frequency information when processing spectrograms. Here, neurons of the hidden layers are arranged into feature maps. Each neuron is connected to a specific fixed portion of the layer's input and all neurons in a single feature map share a set of weights called *kernel* used to convolve the input. Added to convolutional layers, CNNs usually have interleaving pooling layers between the convolutional ones which compress the data while maintaining the overall spatial layout (for instance, calculating the maximum - or average - of neighboring inputs). While, convolutional and pooling layers constitute the feature-learning layers, a classification CNN ends with few fully connected layers forming the classification layers. Alternatively, the calculated features can be fed to an SVM or a KNN to do the classification.

In the following we summarize the characterization of the CNN proposed by [34] and point out the minor differences to the CNN used in our experiments. First, an input point is a 3D-tensor ($3 \times 15 \times 80$) containing 3 magnitude spectrograms with different processing window sizes (23 ms, 46 ms and 93 ms) but same frame rate of 100 fps. The number of logarithmically scaled MEL bands, from 27.5 Hz to 16 kHz, per frame is 80 while the number of frames per data point is 15. A data point is given a label "1" if the middle frame, i.e. the 8th of the 15 frames per data point, is matching an onset. Moreover, fuzziness in training is introduced by labeling the two neighboring data points with "0.25" to account for annotation ambiguity. Note that neighboring data points are 10 ms apart and jointly occur in 14 out of 15 frames. As labeling starts by checking the middle frame, the complete feature sequence per music excerpt is padded with 7 frames at both the start and end, consisting of repetitions of the first and last

Table 5.2: CNN structure [34]

Layer	C_{in}	C_{out}	S_K	S_M
convolution	3	10	7x3	640
max-pool	10	10	1x3	0
convolution	10	20	3x3	1820
max-pool	20	20	1x3	0
fully-connected	1120	256	1	286976
fully-connected	256	1	1	257

frames. No feature normalization is applied as it seemed unnecessary for our experiments.

We used the exact network structure as in [34] and as detailed in Table 5.2, where for each layer, C_{in} denotes the number of input channels, C_{out} is the number of output channels, S_K is the kernel size representing the (time $\times$ frequency) dimensions, and $S_M = (C_{\text{in}} S_K + 1)\, C_{\text{out}}$ is the number of model parameters per layer. For the convolutional and max-pooling layers, channels refer to feature maps while in case of fully-connected layers, channels refer to neurons. A 50 % dropout is applied to the input of each of the fully-connected layers. Convolutional layers use the ReLU activation functions, and the fully-connected layers use the logistic sigmoid function. The training is done on mini-batches of 256 data points for 100 epochs using the adaptive moment estimation (Adam) optimizer [86] with its default parameters and minimizing the binary cross-entropy error. The order of training excerpts is shuffled after each epoch. This was found to result in a performance comparable to what has been reported in [34] when applied on their datasets with 8-fold cross validation and using the same folds.

5.6 Results

Here we compare the performance of the different model training strategies proposed in Section 5.3. Two experiments are run, i.e., separately testing with each of two the real piano datasets MAPS_CL and MAPS_AM, and the resulting F_1 scores are shown in Fig. 5.3 and Fig. 5.4, respectively. Each boxplot

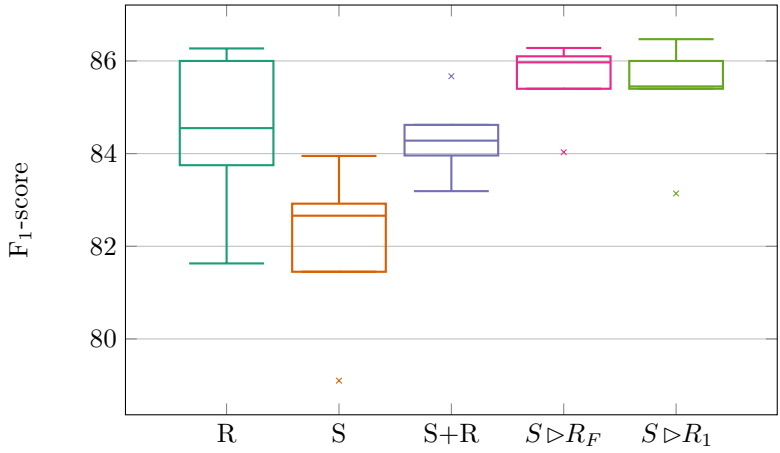


Figure 5.3: F_1 scores for the different model training strategies tested on the MAPS_CL real dataset. Every boxplot is summarizing the 5-fold cross-validation test results.

shows the F_1 distribution among the different folds used per model training strategy (see Section 5.4 for details on the size of the folds for the different datasets).

For both close and ambient recordings, we see that the average performance is better when training on only real data (R) than only semi-synthetic data (S), despite having many more semi-synthetic than real training points. This is not surprising as in the (R) setup, training/validation/testing excerpts originate from the same dataset and are hence much more similar in terms of note onset characteristics. Still, training on the semi-synthetic dataset results in a lower F_1 variance compared to training on the real data, which is an observation that can be extended to all model training strategies involving the semi-synthetic data (i.e., S+R, $S \triangleright R_F$ and $S \triangleright R_1$) in both experiments. Moreover, a better average performance is obtained for almost all strategies in which real and semi-synthetic are combined. The $S \triangleright R_F$ strategy, in which the model was trained to learn note onset features from semi-synthetic excerpts and then retrained on real data, consistently outperforms the other model training strategies in terms of both mean and variance of F_1. This confirms our rationale that a more generic NOD model can be learned from a large semi-synthetic dataset, and that such model can serve as a UOM for onset detection. Comparing the results in Fig. 5.3 and Fig. 5.4, we notice that the onset detection becomes more challenging with ambient microphone recordings, presumably due to the presence of reverberation.

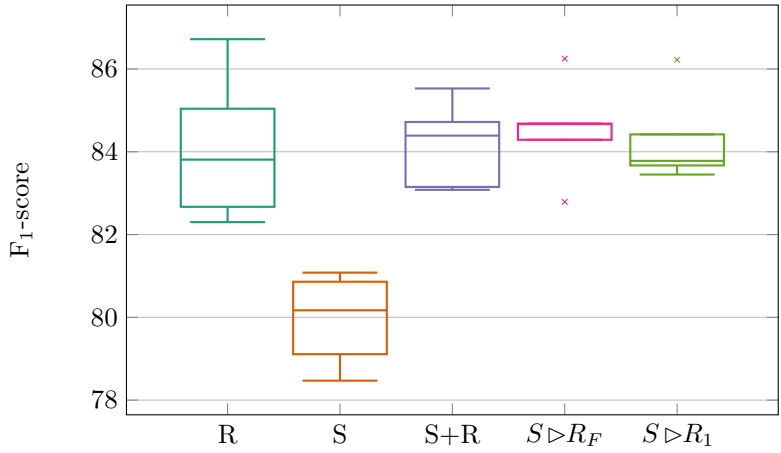


Figure 5.4: F_1 scores for the differentmodel training strategies tested on the MAPS_AM real dataset. Every boxplot is summarizing the 5-fold cross-validation test results.

5.7 Conclusion and future work

We proposed three new model training strategies that allow to augment the scarce datasets for NOD with automatically annotated semi-synthetic data. The best performing strategy consists in training a CNN model on semi-synthetic data for a single instrument type, effectively obtaining a UOM of which the classification layers can then be retrained on a real dataset. We suggest three directions for the further development of this work:

- **Single instruments:** To investigate the effect of further increasing the size and variability of the semi-synthetic dataset on the resulting CNN model performance. Dataset size can be increased by collecting more isolated note recordings and generating more mixes, whereas the variability can be increased by more strongly varying the density of onsets in semi-synthetic mixes, and adding background noise and reverberation.
- **Multiple instruments:** To study the effect of combining different instrument types into a joint semi-synthetic dataset for multi-instrument NOD. In this work, only two extreme cases (single instrument type vs. all possible instruments types) were considered, while a natural way of grouping instrument types would be to consider datasets of instrument families, e.g., winds, pianos, brass and strings.

- **Model complexity:** Due to the automated procedure for creating and annotating semi-synthetic mixes, the dataset size that can be generated in this way is practically unlimited. Having an unlimited amount of training examples opens perspectives to increase the CNN model complexity and enhancing the NOD performance by adding more feature maps or layers.

Chapter 6

Evaluation of note onset detection algorithms using annotations time shift

Annotations Time Shift: A Key Parameter in Evaluating Musical Note Onset Detection Algorithms

Mina Mounir, Peter Karsmakers and Toon van Waterschoot

Published in *Proc. 2019 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '19), New Paltz, NY, USA, Oct. 2019, pp. 21–25.*

© 2019 IEEE. Reprinted, with permission, from:
Mina Mounir, Peter Karsmakers and Toon van Waterschoot, "Annotations time shift: a key parameter in evaluating musical note onset detection algorithms," *Proc. 2019 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '19), New Paltz, NY, USA, Oct. 2019, pp. 21–25.*
Changes include layout, representation, and minor editing aspects.

Abstract

Musical note onset detection is a building component for several MIR related tasks. The ambiguity in the definition of a note onset and the lack of a standard way to annotate onsets, introduce differences in datasets labeling, which in turn makes evaluations of note onset detection algorithms difficult to compare. This paper gives an overview of the parameters influencing the commonly used onset detection evaluation measure, i.e. the F1-score, pointing out a consistently missing parameter which is the overall time shift in annotations. This paper shows how crucial this parameter is in making reported F1-scores comparable among different algorithms and datasets, achieving a more reliable evaluation. As several MIR applications are concerned with the relative location of onsets to each other and not their absolute location, this paper suggests to include the overall time shift as a parameter when evaluating the algorithm performance. Experiments show a strong variability in the reported F1-score and up to 50% increase in the best-case F1-score when varying the overall time shift. Optimizing the time shift turns out to be crucial when training or testing algorithms with datasets that are annotated differently (e.g. manually, automatically, and with different annotators) and especially when using deep learning algorithms.

6.1 Introduction

Detecting the onsets of musical notes is like a hide-and-seek game in which we are trying to chase the starting of musical notes in a piece of music. Usually in those kind of games we have a good idea of what we hide and this moves the whole problem difficulty to the seeking part. For note onset detection this is not the case as literature provides a variety of definitions for a note start. For instance it could be either when a note is triggered or when it is perceived [80]. Considering a musical note as a sequence of a transient followed by a steady-state component [33], an onset is the point chosen to mark the transient [11] or more precisely it should be as close as possible to the transient's start [16]. But again a transient length depends on instrument and playing style and there is no objective way to measure how close is the onset to the transient's start.

Even if note onset detection is an established research problem, it is always capturing researchers' interest. On one hand, there is still quite some room for performance improvement. On the other hand, it plays a core role in a variety of music signal processing (adaptive audio effects [11], music synthesis [70]) and MIR applications (automatic music transcription [12], recommender systems [72] and music fingerprinting/search systems [87], [88], [89]).

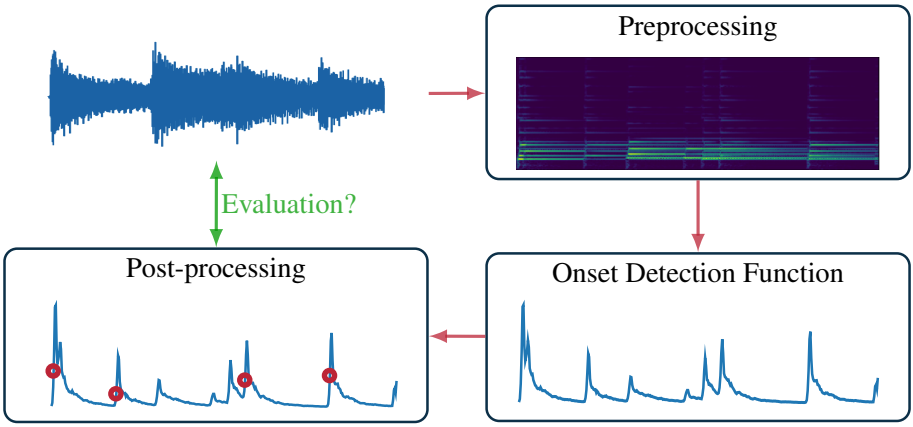


Figure 6.1: General scheme for note onset detection.

Literature is rich with many solutions proposed for detecting note onsets. Each solution starts by deciding on an onset definition which may depend on several factors: application, target instruments, available datasets, labeling method, availability of annotators, etc. The selected definition has a fundamental impact on how the ground-truth annotations are generated which may in turn drastically affect how the detections are evaluated. In many MIR applications like tempo estimation, songs search engines and music synthesis, algorithms make more use of onsets' relative positions rather than their absolute positions. On the other hand, for applications seeking exact onset times, the latter are generally defined with certain tolerance and adapted to the target application.

Before analyzing the performance evaluation, we first summarize the seeking part of the game. It usually follows a certain scheme of three steps [11] illustrated in Fig.6.1. Most of the research is focused on the middle step, trying to come up with a better *Onset Detection Function* (ODF) which is defined as a highly sub-sampled version of the input music signal presenting distinguishable amplitude peaks corresponding to onset locations. Existing ODFs are grouped in two main classes: probabilistic and non-probabilistic. Referring to MIREX results for the last years [61], the best performing state-of-the-art non-probabilistic algorithm was fluctuating between *ComplexFlux* [56] and *SuperFlux* [32] which are based on *LogSpecFlux*(LSF) [37] [44], a method detecting onsets by spectral dissimilarity. Another non-probabilistic algorithm NINOS^2 [33], exploiting spectral sparsity, showed better results than LSF when applied to guitar chord progressions. The same performance is claimed to apply for polyphonic instruments where progressions share more harmonics which challenges LSF. For the probabilistic

ODFs, deep learning algorithms were so far the best performing solutions in MIREX and the convolutional neural network (CNN) by [34] is considered to be the state-of-the-art solution [61].

When evaluating the ODFs using different datasets, one could expect that having a slight shift in the annotations between the training/tuning and testing datasets may result in a misleading performance assessment. This mismatch in labeling is quite common between different datasets due to many reasons, for instance:

- the lack of unified onset definition,
- datasets are labeled by different people and research groups,
- datasets may be of different nature (manually annotated by different annotators, synthesized from symbolic data, natural recording using a Yamaha Disklavier, mixing isolated notes).

In [16] authors suggest a methodology to manually label note onsets in a more consistent way. They also provided a dataset that was used by several researches, but unfortunately is not large enough for training a deep network. On the other hand, in [80], the authors focused on the evaluation parameters and measures. Referring to *Detection Performance* as the most important measure and analyzing the resolution of onset perception, the authors in [80] consider a correct detection to be within a time window of 50 ms around the reference onset time - 25ms on each of the annotation sides - which will be shown not enough when dealing with diverse datasets. In [90], it was shown how a minor labeling mismatch dramatically decreases the evaluation reliability for the melody extraction problem. In that paper, they were mainly considering the mismatch due to framing which is generally smaller than the inter-dataset annotation mismatch.

In this paper we introduce the *overall shift in annotations* parameter providing a means to overcome the datasets' labeling mismatch. We show how this parameter makes the evaluation metric less dependent on datasets and more focused on the performance of detection algorithms. Section 6.2 will give a concise overview of onset detection methods and will explain the parameters influencing the performance evaluation. The experiment environment and datasets will be summarized in Section 6.3. Finally Sections 6.4 and 6.5, will discuss the results and give some suggestions for future work.

6.2 Onset Detection and Evaluation

6.2.1 Onset detection function

Considering the non-probabilistic ODFs, the logarithmic spectral flux (LSF) [37] is given by:

$$LSF(i) \ = \sum_{k=1}^{k=\frac{K}{2}} \ H(|Y_{ik}| - |Y_{i-1,k}|) \ , \tag{6.1}$$

where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function and $|Y_{ik}|$ is the logarithmic magnitude spectrogram for a music signal frame with frame index i, frequency bin k and K is the frame length. NINOS^2 ODF [33] is given by

$$NINOS^2(i) \ = \frac{\|Y_i\|_2^2}{\sqrt[4]{K}\|Y_i\|_4} \ , \tag{6.2}$$

calculating the inverse-spectral sparsity per frame which is high for transients and therefore for onsets. On the other hand the probabilistic CNN is trained on input points labeled with 0 for non-onsets and 1 for onsets. Each point is a 3D tensor formed by concatenating different spectrograms representing a short snippet of the signal. The network's output is treated as an ODF where thresholding is needed afterwards to decide on the output's class. As a consequence of being deep learners, added to the fact of being trained on binary labels, the CNN-ODF is quite different in its nature when compared the non-probabilistic ODFs. This is shown in Fig.6.2 with the dashed lines marking the onsets ground-truth. The CNN-ODF is more precise, i.e. it experiences a magnitude increase in a short window around the onset candidates which may increase evaluation sensitivity to labeling mismatch.

6.2.2 Peak-Picking

In general, the peak-picking starts by smoothing the signal in order to avoid glitches. Then comes either a *fixed* or an *adaptive* thresholding policy depending on the ODF's shape. For non-probabilistic techniques, researchers went to the adaptive choice, calculating a moving average of the ODF and picking the local maxima that are higher than this average with a certain threshold. This is described mathematically by detecting an onset at frame i only if the three following conditions hold:

1. $O(i) = \max_l \ O(i+l), \quad \text{with} \quad l = -\alpha, \ldots, +\beta,$
2. $O(i) \geq \ \frac{1}{a+b+1} \sum_{l=-a}^{+b} O(i+l) + \tau,$

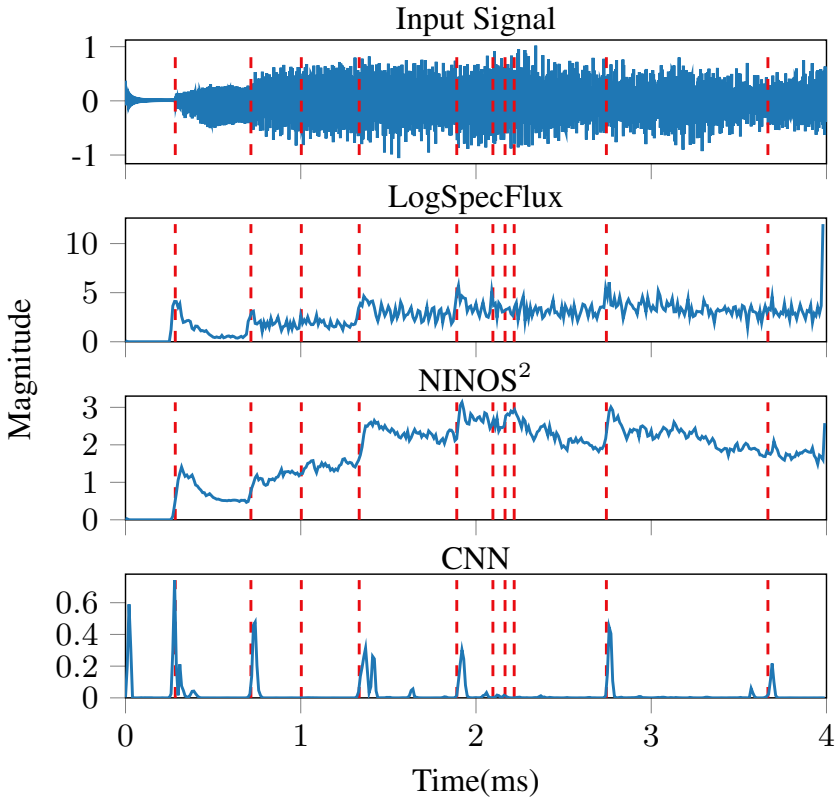


Figure 6.2: Comparison of LogSpecFlux, NINOS^2 and CNN ODFs for a *Cello* excerpt.

3. $i - p > \theta$,

where (α, β) and (a, b) define the maximum and averaging windows respectively and with the last rule restricting the distance between any two onsets i and p to a minimum distance θ. The threshold τ is used when comparing the ODF to the moving average. The parameters α, β, a, b and τ are tuned together or separately for achieving the best performance while θ is heuristically fixed to a distance allowing on average two notes to be distinguishable and hence manually annotated. On the other hand, the CNN produces ODFs where magnitude peaks are centered around candidates' positions and are almost zero elsewhere. Picking all the maxima using only fixed thresholding will do the job reducing all the tunable parameters to the threshold τ. Both peak-picking ways, will produce a list of frame/input-point numbers which represent the detected

onsets. Those frame numbers are usually translated then to time instants before feeding them to the evaluation process.

6.2.3 Evaluation method

For the evaluation we stick to the guidelines mentioned for the MIREX competition [61]. The F1-score is defined as the harmonic average of the *precision* (P) and the *recall* (R) given by:

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{6.3}$$

with the precision being the ratio of correctly detected onsets to the total number of onsets under test, and the recall comparing the amount of correctly detected onsets to the total number of points detected as onsets. When reporting the F1-score, we should differentiate between two cases: best-case and on-average F1-score. Having a limited amount of annotated songs, many researchers are satisfied by sharing the highest achievable F1-score as a proof-of-concept. For this, an F1-score per threshold $\mathrm{F1}(\tau)$ is calculated and averaged over all the test examples, and the highest F1-score corresponding to the optimal threshold τ^* is the reported best-case F1-score. Alternatively, an F1-score is reported for a hold out dataset using the optimal threshold τ^* calculated on the validation set which is similar to MIREX evaluation as the dataset is not public. For a more elaborate evaluation, an on-average F1-score is reported using a k-fold cross-validation [34] which is calculated by averaging the highest F1-score for the different validations sets. Note that in both probabilistic and data-driven algorithms, the F1-score is optimized as a function of the threshold τ only. In general all the remaining parameters (peak-picking and neural network weights) are tuned beforehand on a separate training set. For the purpose of this paper, we will compare the best-case F1-scores as our focus is the evaluation process and not the algorithms' performance.

In the following, it will be shown that the commonly used evaluation window of 50 ms [80] is not enough for dataset mismatch compensation, when the F1-score is optimized considering only the detection threshold τ. Increasing the evaluation window would lead to a worse resolution. That's why we introduce here the overall shift in annotations δ. While searching for the best threshold, this parameter can be used to find the best match between annotations and detections from the timing point of view. This is done by shifting the annotations for the whole dataset to the right (or left) by adding a small offset δ and searching for the best overlap corresponding to the highest F1-score. In other words, F1-score denoted by $\mathrm{F1}(\tau,\delta)$ is optimized over two parameters: detection threshold τ and overall shift in annotations δ. In our

opinion, this doesn't weaken the performance measure. On the contrary, it makes the performance assessment less dependent on the labeling definition or the used method or persons asked to annotate the dataset.

6.3 Experimental evaluation

6.3.1 Datasets

In order to assess the effect of the overall shift in annotations δ on the F1-score, we used four datasets to which we give the names: MDS, SDS, MAPS_CL, MAPS_AM. The MDS dataset is the one used to evaluate the CNN for onset detection [45]. It is a manually annotated dataset containing audio excerpts from various sources. A 20% portion of this dataset is used for training the network and tuning the peak-picking parameters - except τ - while the remaining examples added to the remaining datasets are used to form four different testing use-cases. The SDS is an automatically annotated dataset [33], containing 138 audio examples. Each example is a mix of 70 notes annotated beforehand - when the notes were isolated - depending on an energy measure. Finally, the MAPS_CL and MAPS_AM are part of the MAPS[1] dataset, specifically the music pieces portion of ENSTDkCl and ENSTDkAm. 'Cl' and 'Am' acronyms refer to *close* and *ambient* way of recordings specifying the distance to the microphone. Those two datasets were recorded using a Yamaha Disklavier and annotations are generated automatically. Table 6.1 gives an overview of the used training and test sets in terms of number of music pieces, onsets and the corresponding number of input points fed to the CNN.

[1] *Midi Aligned Piano Sounds* dataset - freely available under Creative Commons license

Table 6.1: Datasets summary

Name	Files #	Onsets #	Points #
MDS_train	77	5979	150K
MDS_test	204	18540	400K
SDS	138	9660	650K
MAPS_CL	30	76364	800K
MAPS_AM	30	77988	800K

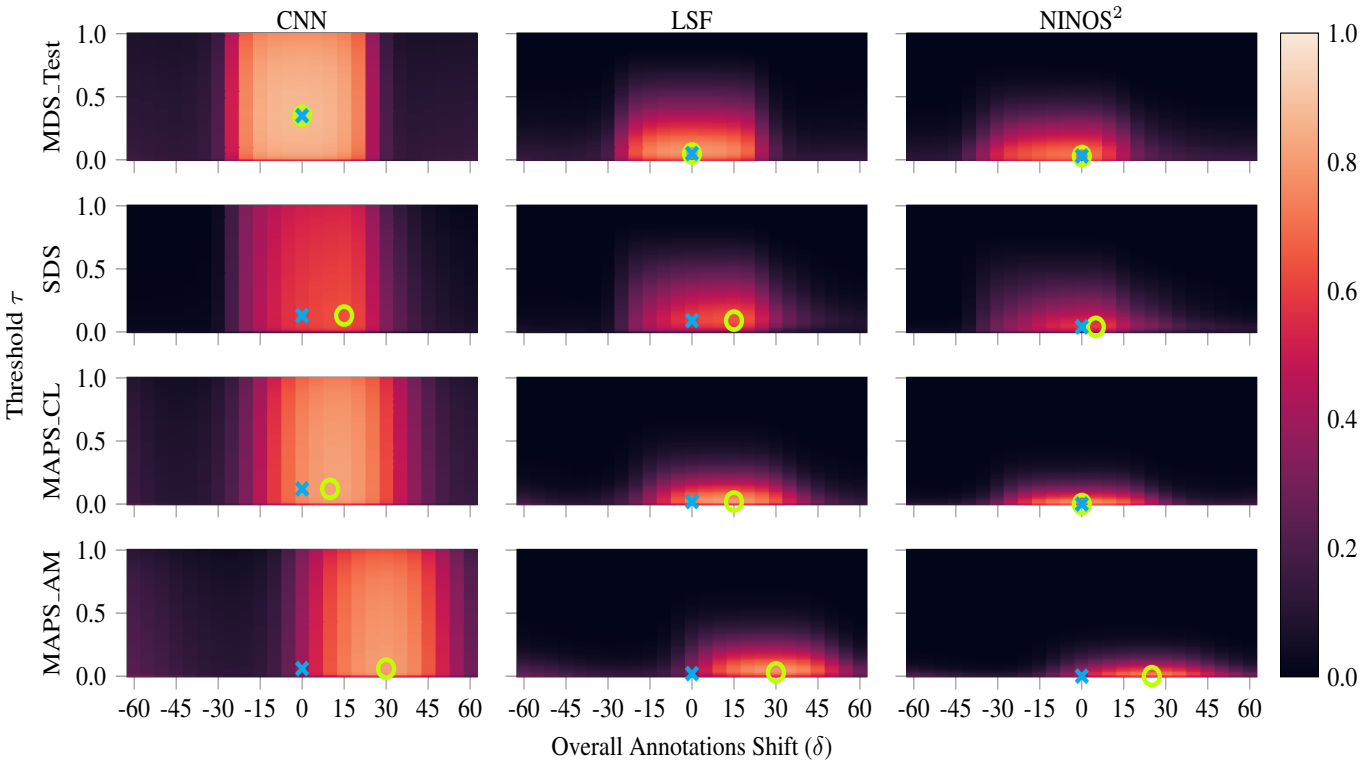


Figure 6.3: F1-score as a function of threshold τ (vertical axis) and shift δ (horizontal axis), averaged per test set. *Circles* and *crosses* respectively represent the best F1-score average with and without considering the overall shift in annotations δ.

6.3.2 Experiment setup and parameters

For the CNN we used the same characterization as in [34] with minor differences pointed out here. First, an input point is a 3D-tensor (3x15x80) holding 3 magnitude spectrograms with different processing window sizes (23 ms, 46 ms and 93 ms) but same frame rate of 10 ms. The number of logarithmically scaled MEL bands per frame is 80 while 15 is the number of frames per point. A point is given a label '1' if the middle frame is matching an onset and '0.25' for its neighboring points to handle annotations ambiguity. While 50% dropout in training is maintained, no feature normalization is applied which was not necessary for the experiment. Convolutional layers use the ReLU activation functions pushing the CNN to learn spectral differences like the state-of-the-art non-probabilistic ODF's, and the fully-connected layers use the logistic sigmoid. The training is done on mini-batches of 256 points for 150 epochs using the Adam optimizer [86], minimizing the cross-entropy error. The order of training examples is shuffled after each epoch. This was found to achieve comparable results to the ones reported in [34] when applied on the same datasets with 8-fold cross validation setup and using the same folds.

For the pre-processing, peak-picking and evaluation parameters we stick to the ones used in [34] and [33]. Frames are smoothed with a 50 ms hamming window. For the peak-picking we set θ, α and β to 30 ms, a to 100 ms and b to 70 ms. The evaluation window is kept 50 ms around the ground-truth. Only the probabilistic ODFs were normalized in the range $[0, 1]$ before applying the peak-picking as the algorithms' online capabilities are not our concern in this study.

6.4 Results

In order to study the effect of including the overall shift in annotations δ, we calculated a grid of the average F1-score per dataset as a function of the threshold $\tau \in [0, 1]$ with a step 0.01 vs the shift $\delta \in [-60, 60]$ ms with a step of 5 ms.

Figure 6.3 shows the threshold-vs-shift F1-score grids for the four test sets using the three ODFs. While the F1-score is less sensitive to threshold variations, it is remarkable how the regions marking the F1-score peaks are not aligned for the different datasets. The F1-score peak values are centered around $\delta = 0$ when the training and testing examples are coming from the same dataset MDS while being always shifted for the other datasets. By comparing the best $\mathrm{F1}(\delta, \tau)^*$ to the zero-shift $\mathrm{F1}(\delta = 0, \tau^*)$ in Table 6.2, we can see that it is consistently registering a higher F1-score upper bound for the synthetic and

disklavier datasets. For instance, neglecting the overall annotations shift for the MAPS_AM dataset reports a best-case performance of more than 50% less than the actual one due to labeling mismatch between the datasets. Note that the shifts $\delta \in [5, 30]$ allowing for a better assessment are calculated on top of the evaluation window, which means a labeling mismatch of up to $30 + 50/2 = 55$ ms which would need an evaluation window of 110 ms to handle it using the methodology in [80]. Different δ values were needed to attain the best scores for the different algorithms. This tells how the different ODFs reply differently to the *seeking* question which confirms the need for a better evaluation independent from the shift.

Table 6.2: Comparison of best-case F1-score with and without *Overall Shift in Annotations δ Parameter*

		Zero-Shift		Considering the Shift		
	Fold	F1(%)	τ	F1(%)	δ(ms)	τ
CNN	MDS_test	86.7	0.36	86.7	0	0.36
	SDS	59.5	0.14	63.1	15	0.14
	MAPS_CL	75.9	0.13	81.1	10	0.13
	MAPS_AM	39.6	0.05	80.0	30	0.07
LSF	MDS_test	78.4	0.60	78.4	0	0.60
	SDS	59.5	0.10	64.7	15	0.10
	MAPS_CL	72.8	0.03	79.3	15	0.03
	MAPS_AM	36.6	0.03	78.3	30	0.04
NINOS^2	MDS_test	70.2	0.04	70.2	0	0.04
	SDS	57.5	0.05	59.3	5	0.05
	MAPS_CL	76.2	0.01	76.2	0	0.01
	MAPS_AM	45.8	0.01	72.1	25	0.01

6.5 Conclusion & Future work

This paper argues that the use of an overall shift in annotations parameter δ improves the onset detection evaluation reliability coping with the labeling mismatch between datasets that are annotated differently. δ is crucial for a sound performance assessment making the evaluation process less dependent on the annotation method and thereby more focused on the assessment of the detection algorithm. When reporting the best-case F1-score, we suggest providing the full $F1(\delta, \tau)$ grid. For the on-average F1-score, a possibility is to search for the optimal $(\delta, \tau)^*$ using a validation set taken out from the same dataset as the test set. In the future, we would like to make use of the shift mismatch results to align the examples labeling when mixing together different datasets for training. Moreover, this work can be extended to any MIR time-dependent task.

6.6 Additional results

For completeness, the results for identifying note onsets based on spectral sparsity ($\text{INOS}^2(\ell_1)$), the computationally cheap variation of the spectral sparsity note onset detection (NOD) algorithms introduced in Chapter 4, are presented in Table 6.3. On the other hand, the results for normalised identification of note onset based on spectral sparsity ($\text{NINOS}^2(\ell_2\ell_4)$) with the new normalization procedure defined in (4.12) is not reported here, as the presented work uses a fixed processing window size which leads to the same results for $\text{NINOS}^2(\ell_2\ell_4)$ as those reported in Table 6.2 for normalised identification of note onset based on spectral sparsity (NINOS^2).

Again, we can see that the performance of identifying note onsets based on spectral sparsity (INOS^2) is altered when considering the overall shift in annotations parameter δ and especially with the MAPS_AM dataset. Originating from the same family of NOD features, it is not surprising that the best performance is obtained with the same δ values as for NINOS^2. The overall performance of INOS^2 shows an improvement over NINOS^2 when applied on the different datasets, while staying slightly behind the Logarithmic Spectral Flux (LSF) feature performance. This may seem contradicting with the results reported earlier in Chapter 4, but it is possibly happening because of two reasons:

- The evaluation datasets are different (including the semi-synthetic data).
- The features' hyper parameters were not tuned for optimal performance per feature as the main objective of this chapter is to show the effect of the overall shift parameter on the performance of the different features. For instance, a coefficients subset percentage γ=95.5 is used for the pre-processing of (N)INOS^2 as what was reported optimal for the experiment of Chapter 4.

In Table 6.4, we show the results when running the overall shift experiment on the *strings* instruments portion of the *PR8* dataset used in the evaluation of Chapter 4 and using the respective optimal hyper parameters for LSF, NINOS^2 and INOS^2. The results shows the same trend as in Section 4.4.5 where (N)INOS^2 performs better than LSF and Convolutional Neural Network (CNN). It is interesting to see how CNN's performance is quite behind the non-data-driven features. This is because the CNN is trained on MDS_train which is a dataset of different nature (manually annotated and containing various types of music, e.g., jazz, concerts, classic, individual instruments, etc).

Table 6.3: Best-case F1-score with and without *Overall Shift in Annotations* δ *Parameter* for INOS$^2(\ell_1)$.

		Zero-Shift		Considering the Shift		
	Fold	F1(%)	τ	F1(%)	δ(ms)	τ
INOS$^2(\ell_1)$	MDS_test	70.8	0.04	70.8	0	0.04
	SDS	57.4	0.05	59.2	5	0.05
	MAPS_CL	77.8	0.01	77.8	0	0.01
	MAPS_AM	48.8	0.01	74.4	25	0.01

Table 6.4: Comparison of best-case F1-score with and without *Overall Shift in Annotations* δ *Parameter* when evaluated on the *strings* instruments of the *PR8* dataset of Chapter 4.

Model/Feature	F1(%)	δ(ms)	τ
CNN	43.0	20	0.01
LSF	47.8	25	0.10
NINOS$^2(\ell_2\ell_4)$	48.9	35	0.04
INOS^2 (ℓ_1)	48.3	35	0.04

Part III

Howling Detection

Chapter 7

A large and diversified dataset for howling detection

7.1 Introduction

In most of the scientific literature on howling detection, results are reported for small howling datasets or even just for a few examples, serving as a proof of concept rather than as a thorough performance evaluation [15]. In some datasets used in literature, audio signals containing howling is recorded and then manually labeled [38, 91], whereas in other datasets, howling is simulated through the superposition of sinusoids (pure tones) to clean audio signals aiming to imitate the specific narrowband characteristics of howling [38, 48]. Both approaches have their advantages and drawbacks. Simulated datasets using superposition of sinusoids are easy to generate and annotate, yet such synthetic howling datasets do not guarantee that all characteristic features of real howling are adequately represented, in particular early howling and ringing, which refers to the early phase of a non-continuous howling artefact. Recorded datasets on the other hand are more realistic but require manual annotation, which is time-consuming and subjective and hence implicitly introduces dependencies between training and testing datasets.

In this work, with the aim of achieving a thorough evaluation of the novel howling detection feature introduced in Chapter 8 and comparing it with existing features, a new database of howling-corrupted audio signals is created. Unlike other dataset generation approaches proposed in literature, howling is created by simulating an electroacoustic closed-loop system. This approach

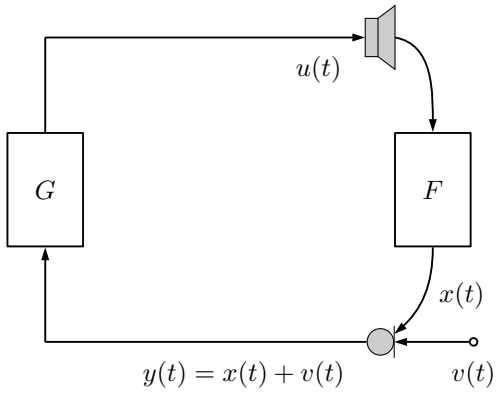


Figure 7.1: Electroacoustic closed-loop (feedback) system.

combines the advantages of the aforementioned methods while circumventing their shortcomings, *i.e.*, the simulated signals exhibit a realistic howling or ringing component while allowing for an automatic annotation.

7.2 Dataset generation

An example of said closed-loop system is shown in Fig. 7.1. Let t be the discrete time index and q the discrete time shift operator, *i.e.*, $q^{-k}u(t) = u(t-k)$. The signal $y(t)$ captured by the microphone does not only consist of the source signal $v(t)$ but also contains the feedback signal $x(t)$, which is in turn the result of the loudspeaker signal $u(t)$ after traversing the acoustic feedback path. The transfer functions $G(q)$ and $F(q)$ are respectively representing the feed-forward and feedback paths, the characteristics of which have a direct impact on the simulated howling generation. A more detailed explanation of the closed-loop system and its stability properties are left to Chapter 8.

Eight acoustic impulse responses (AIRs) simulating the feedback path F are selected from an AIR database [92] and each AIR is either truncated or padded to a length of 1 s at a sampling frequency $f_s = 16$ kHz. The eight selected AIRs are chosen in order to achieve sufficient variability in the frequencies of the simulated howling signals. All the selected AIRs are normalized such that the maximum stable gain (MSG) condition of the closed-loop system is achieved when providing a forward path gain G of 10 dB.

The forward path of the closed-loop system is chosen to be a time-varying

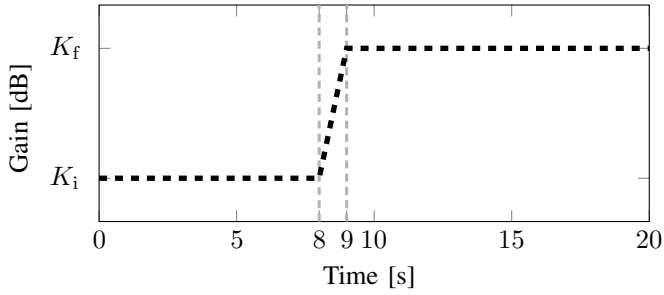


Figure 7.2: Gain profile of the forward path transfer function.

broadband gain with the gain profile shown in Fig. 7.2. The gain is fixed to $K_\mathrm{i} = \mathrm{MSG} - 6\,\mathrm{dB}$ for the first 8 s, linearly increased to $K_\mathrm{f} = \mathrm{MSG}$ in the following 1 s, and fixed to K_f for the remaining 11 s of the simulation.

Both speech and music signals are used as the source signal $v(t)$. The speech signals are 20 s excerpts, sampled at $f_s = 16\,\mathrm{kHz}$, of recorded male and female speech from an audiobook database available in four different languages (Chinese, English, Dutch, and Russian) for a total of eight source signals [93]. The music signals are 20 s excerpts, sampled at $f_s = 16\,\mathrm{kHz}$, of seven music pieces from different genres (jazz, opera, etc...) [83, 94]. Gaussian white noise is added to the source signal $v(t)$ with a fixed signal-to-noise ratio (SNR) of 40 dB in all the recordings. All the source signals are normalized to have the same root mean square (RMS) value.

Each of the eight speech and seven music source signals are fed to the closed-loop system shown in Fig. 7.1, with each one of the eight AIRs as F, with the aforementioned gain profile followed by a 20 ms delay as G. After the gain, a hard clipping saturation is applied with clipping level one to roughly model the loudspeaker's nonlinear behavior. This yields a total of 120 simulated signals. The filtering operation of F on $u(t)$ was implemented by means of the overlap-and-save algorithm [95]. For each signal in the resulting dataset, an annotation file is automatically generated which contains the various information about the howling simulation, *e.g.*, the MSG frequency (*i.e.*, the frequency at which the loop gain is 0 dB when the forward path gain is equal to the MSG), gain profile, AIR, source signal file, etc. An example of one such annotation file is given in Table 7.1.

Table 7.1: A howling annotation file example.

Parameter	Value
Audio filename	music_in_01.wav
AIR filename	AIR_007.wav
Start gain	4 dB
End gain	10 dB
Sampling Frequency	16 kHz
Gain transition start time	8 s
Gain transition length	1 s
Audio length	20 s
MSG frequency	579.8 Hz

7.3 Problematic samples and solution

Upon the creation of the database, we noticed that some of the simulated howling-corrupted signals were problematic and were deemed not suitable for the subsequent howling detection evaluation procedure. The main problems encountered were the following (see Fig. 7.3):

- Lack of howling in some of the signals (which occurred only in the case of music signals).
- Signals exhibiting howling (and/or feedback-related artefacts) at multiple frequencies. This effect was observed and loosely classified into two different cases:

    1. Howling at multiple non-harmonically-related frequencies, caused by the presence of multiple frequencies at which the loop gain is (around) 0 dB when the forward path gain is equal to the MSG. Depending on the behavior of the loop gain as a function of frequency, the multiple howling frequencies can either be well spaced, giving rise to a set of clearly distinct howling or ringing frequencies, or nearby (often due to broadband peaks in the loop gain) giving rise to multiple howling

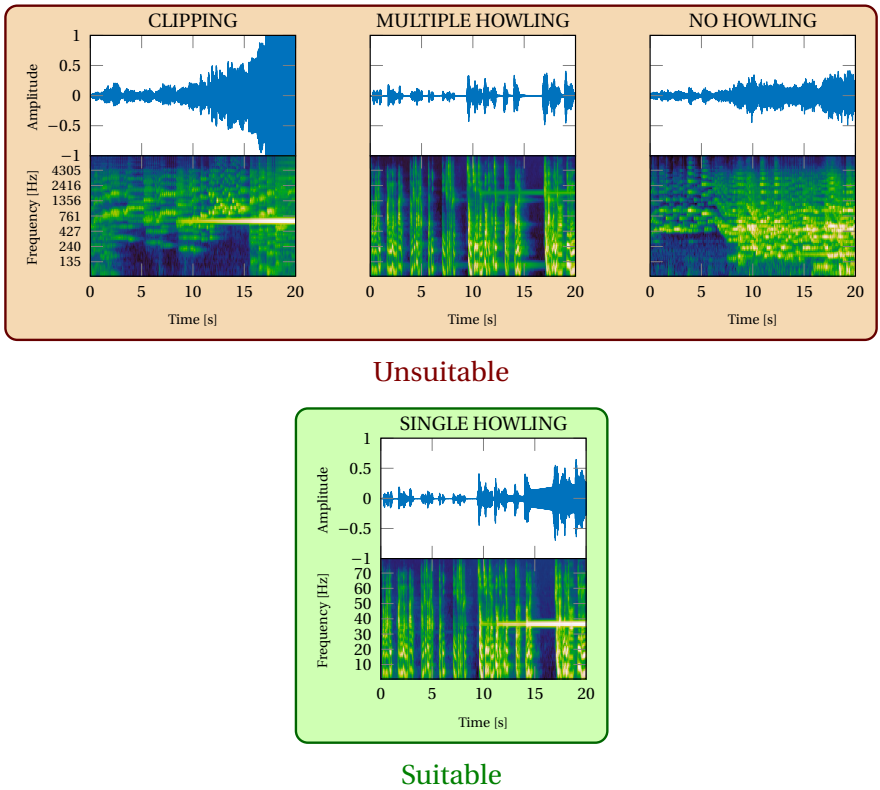


Figure 7.3: Example of unsuitable and suitable signals for the dataset.

or ringing frequencies in the same frequency band which are often perceived as a single howling frequency.

2. Howling at multiple harmonically-related frequencies, caused by the large signal amplitude at the original howling frequency exceeding the hard clipping saturation level.

The identified causes leading to the creation of these problematic signals are the following:

- Estimation errors on the MSG and the corresponding howling frequency, due to the limited frequency resolution used in the MSG calculation, may have played a role in all three types of problems encountered.

- The large dynamic range of music signals makes the normalization by means of the RMS less effective. For example, an RMS-normalized signal may coincidentally exhibit larger amplitudes in the beginning (*i.e.*, before the transition in the gain profile in Fig. 7.2) and smaller amplitudes afterwards, due to which howling may not be (perceptibly) triggered.

Therefore, the dataset is pruned in order to remove the signals deemed not suitable. In order to do so, a procedure to remove unsuitable signals has been devised and independently carried out by three experts on all 120 simulated signals in the dataset. The procedure starts by listening to the signals and visually inspecting their spectrograms, after which the expert labels each signal as either "suitable" (visible/audible howling at a single frequency) or "unsuitable" (no visible/audible howling, or visible/audible howling at multiple non-harmonically-related frequencies or at multiple harmonically-related frequencies). Figure 7.3 shows examples of suitable and unsuitable simulated signals. Finally, the pruning is based on the combination of the individual expert labels as follows:

1. If **any** expert identified the signal as having multiple howling frequencies (harmonically-related or not), the signal is excluded from the dataset. Else, go to 2.
2. If **at least 2** experts identified that no howling occurs, the signal is excluded from the dataset. Else, go to 3.
3. The signal is kept in the dataset.

In this way, the different problems are effectively handled with different severity, *i.e.*, it is judged more important to exclude signals having multiple howling frequencies than signals having no howling. The resulting dataset after pruning contains 28 music files and 30 speech files where each source signal (music or speech) and each AIR is used at least in one of the simulated signals. This constraint guarantees to obtain a dataset covering a wide range of source signal types and AIRs, which was indeed one of our original objectives.

7.4 Conclusion

This chapter starts by discussing the limitations of the available datasets used for evaluating howling detection (HD) algorithms. They are either too simplistic and not capable of simulating ringing and early howling or subjective and small-sized. Differently from the approaches proposed in literature, Chapter 7

suggests an automatically annotated howling-corrupted dataset by simulating an electroacoustic closed-loop system. It details the generation method for the samples together with their labeling and the used source files (music and speech) and AIRs. Chapter 7 discussed the presence of few problematic samples in the generated dataset, having either multiple howling or no howling at all. It analyzed the possible causes and suggested a pruning procedure – conducted by 3 experts – as a solution. The resulting dataset is a diversified automatically-annotated howling-corrupted dataset covering different input types and large howling-frequency range[1].

As a future step, it should be investigated if these howling generation issues (multiple and no-howling) can be mitigated during the generation without needing a following pruning step. On the other hand, extending the labeling to multiple howling can be beneficial for evaluating multiple howling detection.

[1] available for download at `https://limo.libis.be/primo-explore/fulldisplay?docid=LIRIAS3119197&context=L&vid=Lirias&search_scope=Lirias&tab=default_tab&lang=en_US`

Chapter 8

Howling detection based on a spectral sparsity measure

Robust and early howling detection based on a sparsity measure

Mina Mounir, Giuliano Bernardi and Toon van Waterschoot

submitted for publication.

Abstract

Despite recent advances in audio technology, acoustic feedback remains a problem encountered in many sound reinforcement applications, ranging from public address (PA) systems to hearing aids (HA). Acoustic feedback occurs due to the acoustic coupling between a loudspeaker and microphone, creating a closed-loop system that may become unstable and produce an acoustic artifact referred to as *howling*. One solution to the acoustic feedback problem, known as notch-filter-based howling suppression (NHS), consists in detecting and suppressing howling components hence stabilizing the closed-loop system and removing audible howling artefacts. The key component of any NHS method is howling detection (HD), which is typically based on the calculation of temporal and/or spectral features that allow to discriminate howling from desired audio signal components. In this paper, three contributions to HD research are presented. Firstly, we propose a novel howling detection feature, coined as NINOS^2-Transposed (NINOS^2-T), that exploits the particular time-frequency structure of a howling artefact. The NINOS^2-T feature is shown to outperform common state-of-the-art HD features, to be more robust to detection threshold variations, and to allow for the detection of early howling and ringing by discarding the often used concept of howling candidates selection. Secondly, a new annotated dataset for HD research is introduced which is significantly larger and more diverse than existing datasets containing realistic howling artefacts. Thirdly, a new HD performance evaluation procedure is proposed that is suitable when using HD features that do not rely on a howling candidates selection. This procedure opens the door for the evaluation of early howling and ringing detection performance and can handle the high class imbalance inherent in the HD problem by using precision-recall (PR) instead of receiver operating characteristic (ROC) curves.

8.1 Introduction

During a nice concert or while attending a lecture, many of us experienced this whistling sound that rapidly grows too loud, forcing us to plug our ears hoping this will not last too long. This extremely annoying sound affecting the audio quality – both to the listener and to the performer – is called *howling*. It is the most characteristic artifact of the acoustic feedback problem [14], which occurs due to an unwanted acoustic coupling between a loudspeaker and microphone in a sound reinforcement system. Public address (PA) systems and hearing aids (HA) are two applications where howling occurs frequently even when designed to avoid a direct acoustic coupling. This is because the indirect acoustic coupling

caused by reflections on boundaries, objects or subjects in the environment, is unavoidable. As will be explained in detail in Section 8.2, howling results from the instability of the closed-loop system originating from the acoustic coupling. In stable closed-loop systems operating close to instability , another feedback artefact known as ringing may occur. In contrast to howling, ringing does not persist yet it may be equally detrimental to audio quality.

Manual solutions to overcome these annoying feedback artefacts include changing the microphone position or lowering its gain. Automatic solutions, jointly denoted as acoustic feedback control (AFC) methods, exist in many flavours and aim to solve the feedback problem either completely by removing the acoustic coupling, or partially by detecting and eliminating the artefacts. One category of AFC methods are the so-called automatic gain reduction methods, which are widespread in sound reinforcement systems [15], mimicking the immediate action an audio engineer would undertake to remove howling as soon as it occurs. The automatic gain reduction methods can further be grouped depending on the width of the frequency band they are attenuating [14]. Automatic gain control (AGC) methods apply a broadband gain reduction, automatic equalization (AEQ) methods attenuate critical subbands, and notch-filter-based howling suppression (NHS) methods apply a narrowband gain reduction around the critical frequencies, *i.e.*, frequencies where howling/ringing occurs, possibly resulting in the lowest audio signal distortion. NHS methods use spectral and/or temporal features for identifying critical frequencies. Most gain reduction methods are reactive, *i.e.*, howling is detected before being suppressed. This makes the performance evaluation of NHS methods not only concerned about the detection precision, but also about detection speed and audibility of howling before detection.

The fast and accurate detection of howling occurrence is therefore the main challenge in the design of an effective NHS method. Adopting a classical detection theory viewpoint, the howling detection (HD) problem can be cast as a binary hypothesis testing problem (howling vs. no howling) which is typically solved by comparing some signal feature to a detection threshold [15]. A detailed overview of signal features that have been proposed for HD will be provided in Section 8.2. A common property of these features, which will be challenged in this paper, is that howling can only be detected after it has manifested itself as a peak in the microphone signal magnitude spectrum. These features are therefore not calculated for the entire frequency range of interest but only for a few candidate frequencies that are peak-picked from the microphone signal magnitude spectrum. This presumption implicitly rules out the detection of early howling and ringing components, since these artefacts exhibit too little energy to be peak-picked from the microphone signal magnitude spectrum.

A second issue with existing NHS methods, most of which are described in

patents or proof-of-concept-oriented papers, is that their HD performance evaluation is often carried out using few examples and moreover the algorithm parameters, *e.g.*, the detection threshold, are tightly tailored to these examples [15]. More extensive datasets used for HD performance evaluation [91], [38], [48] are either synthetically generated by adding pure tones (sinusoids) to audio signals or based on real howling recordings lacking details on the annotation procedure. While the synthetic datasets are not guaranteed to be representative of realistic howling phenomena and for sure not of early howling and ringing, the recorded howling datasets are manually annotated and therefore relatively scarce. Moreover, the subjective nature of manual annotation implicitly introduces dependencies between training and testing datasets.

A final challenge in NHS research that is considered in this paper concerns the effective performance evaluation of HD features. Due to the fact that existing HD features are only used to discriminate between howling and desired audio signal components for a few candidate frequencies that are peak-picked from the microphone signal magnitude spectrum, their reported performance evaluation may be too optimistic. That is, howling can only be detected when it is included in the set of candidate howling frequencies which implies the howling component exhibits relatively large energy and is thus easy to detect. The more challenging low-energy howling and ringing components are usually not included in the set of candidate howling frequencies and are therefore not counted in the performance evaluation. Moreover, HD performance evaluation is usually carried out with the receiver operating characteristic (ROC) curve and the area under the ROC curve (ROC-AUC) measure, which assess how well an algorithm detects howling while avoiding false alarms. Despite being a powerful detection performance evaluation tool, the ROC has been proven to perform poorly with unbalanced datasets [59] which is generally the case for howling datasets as these contain many more realizations of non-howling than of howling.

In this paper, each of the three issues discussed above is addressed with a novel contribution. A new HD feature named NINOS^2-Transposed (NINOS^2-T) is proposed, which is a modified version of the normalised identification of note onset based on spectral sparsity (NINOS^2) feature previously developed for the musical note onset detection problem [33]. This feature exploits the particular time-frequency structure of a howling component and does not rely on a preselection of candidate howling frequencies. The performance of the proposed and state-of-the-art HD features is evaluated on a novel automatically annotated dataset of howling-corrupted speech and music signals, which is larger and more diverse than existing datasets. Finally, the performance evaluation is designed such that also the early howling and ringing detection performance can be assessed, by considering all frequency bins as candidate howling frequencies.

Precision-recall (PR) curves and area under the PR curve (PR-AUC) and F1-score measures are used instead of the ROC evaluation tool to compare the different features, due to their suitability for skewed datasets which are indeed more representative of practical HD use cases.

The rest of this paper is organized as follows. In Section 8.2, we briefly introduce the acoustic feedback problem, summarize the NHS solution scheme and introduce the different state-of-the-art HD features. In Section 8.3, the proposed HD feature NINOS^2-T is derived from its NINOS^2 counterpart that was previously developed for note onset detection. In Section 8.4, we describe how the dataset used to evaluate the different HD features was created. In Section 8.5, we describe the new HD performance evaluation procedure and illustrate how it makes the comparison of HD features more reliable and practicality oriented. In Section 8.6, we provide simulation results, and finally the conclusions are drawn in Section 8.7 with suggestions for future work.

8.2 Problem statement and existing solutions

8.2.1 Acoustic feedback problem

In order to understand the conditions leading to howling, we first study how the acoustic feedback problem is modeled with a closed-loop system model as shown in Fig. 8.1 considering a single channel with one microphone and one loudspeaker. Using t as the discrete time index and q as the discrete-time shift operator, *i.e.*, $q^{-k}u(t) = u(t-k)$, the different signals and transfer functions can be described as follows:

- $v(t)$ represents the source signal acquired by the microphone (*e.g.*, speech or music);
- $y(t)$ represents the microphone signal fed to the forward path transfer function $G(q)$ of the closed-loop system;
- $u(t)$ represents the loudspeaker signal, *i.e.*, the output of $G(q)$, played back by the loudspeaker;
- $x(t)$ represents the feedback signal, *i.e.*, the output of the feedback path transfer function $F(q)$ causing the unwanted coupling between loudspeaker and microphone.

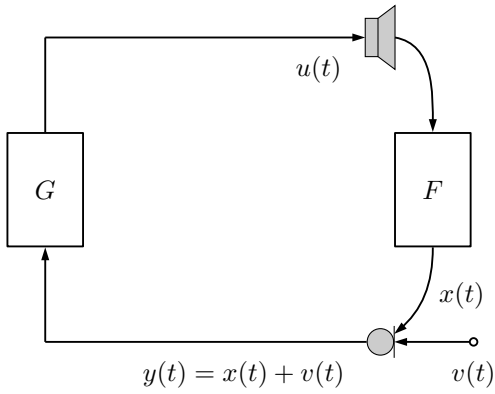


Figure 8.1: Closed-loop (feedback) system model.

The closed-loop frequency response is given by

$$\frac{U(\omega,t)}{V(\omega,t)} = \frac{G(\omega,t)}{1 - G(\omega,t)F(\omega,t)} \ , \tag{8.1}$$

where $U(\omega,t)$ and $V(\omega,t)$ represent the short-term frequency spectra of the loudspeaker and source signal, $G(\omega,t)$ and $F(\omega,t)$ represent the short-term frequency responses of the forward and feedback paths and $\omega \in [0, 2\pi]$ is the radial frequency variable. The forward and feedback paths are assumed to be slowly time-varying. A closed-loop system exhibits instability at a frequency $\omega = 2\pi\,(f/fs)$ when satisfying the conditions on the *loop gain* and *loop phase*, formulated by the Nyquist stability criterion:

$$|G(\omega,t)F(\omega,t)| \geq 1, \tag{8.2}$$

$$\angle G(\omega,t)F(\omega,t) = n2\pi, \quad n \in \mathbb{Z}. \tag{8.3}$$

Under these conditions, if the system is excited at the frequency ω, an oscillation due to instability occurs and may become perceivable in the form of howling.

The maximum stable gain (MSG) is defined as the maximum broadband gain that can be applied in the closed-loop system forward path without rendering the system unstable. Under the assumption that the forward path merely consists of a broadband and possibly time-varying gain (referring to [14] for a treatment of the more general case), the MSG is given by:

$$MSG(t)\,[\mathrm{dB}] = -20\log_{10}\big(\max_{\omega\in\mathcal{W}}|F(\omega,t)|\big) \tag{8.4}$$

where $\mathcal{W}$ is the set of radial frequencies satisfying the loop phase condition in eq. (8.3).

The spectrogram in Fig. 8.2 shows how howling occurs when the two conditions in the Nyquist stability criterion are satisfied. In a spectrogram representation, howling can usually be visually identified – perhaps slightly later than when it actually starts – as a high-energy horizontal line, *i.e.*, a persistent, narrowband and often high-frequency signal component. Out of the three frequencies in this example that satisfy the loop gain criterion (8.2), *i.e.*, the loop gain exceeds the 0 dB line, only one frequency (around 1.6 kHz, indicated by the red dashed line) also satisfies the loop phase criterion (8.3), *i.e.*, the unwrapped loop phase crosses the zero radians line. A frequency that satisfies the loop phase criterion (8.3) but not the loop gain criterion (8.2), may still give rise to another acoustic feedback artefact called *ringing*. This can be noticed in Fig. 8.2 at a frequency around 2.6 kHz (indicated by the orange dashed line) where the unwrapped loop phase is zero while the gain is around −3 dB.

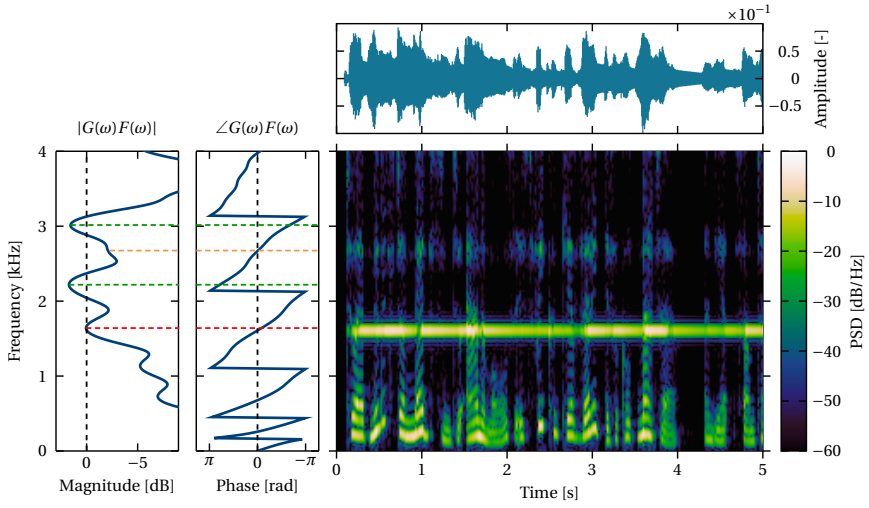


Figure 8.2: Nyquist stability criterion example.

8.2.2 Notch-filter-based howling suppression (NHS)

NHS is a two stage solution: first, howling is detected and its frequency and magnitude estimated, then – if howling was detected – a notch filter is designed to suppress it. Figure 8.3 shows how NHS can be integrated in the closed-loop system. A HD block detects howling in the microphone signal $y(t)$ and provides

a set of notch filter design parameters $\mathcal{D}_H(t)$ to the filter block $H(\omega, t)$, which represents a bank of adjustable notch filters activated and deactivated when needed for howling suppression.

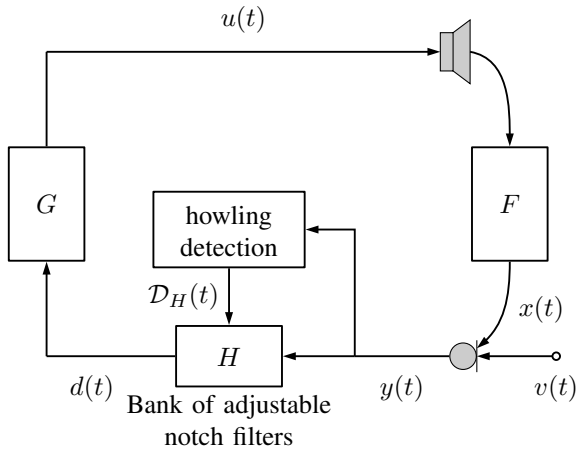


Figure 8.3: NHS solution scheme.

The HD block constitutes the most critical component of the NHS solution scheme [14]. The large majority of HD methods proposed in literature [15], [91], [38], follows the processing scheme presented in Fig. 8.4, see [15] for a detailed discussion. The microphone signal $y(t)$ is divided into overlapping frames with M samples each and a hop size equal to P,

$$\mathbf{y}(t) = \begin{bmatrix} y(t+P-M) & \dots & y(t+P-1) \end{bmatrix}^T. \tag{8.5}$$

Each frame is then transformed to the frequency domain using the short-time Fourier transform (STFT) after undergoing a windowing function to avoid spectral leakage, *i.e.*,

$$Y(\omega_k, t) = \sum_{n=0}^{M-1} w(t_n) y(t_n) e^{-j\omega_k t_n}, \quad k = 0, \dots, M-1, \tag{8.6}$$

$$\mathbf{Y}(t) = \begin{bmatrix} Y(\omega_0, t) & \dots & Y(\omega_{M-1}, t) \end{bmatrix}^T, \tag{8.7}$$

with the angular frequency $\omega_k \triangleq 2\pi k/M$, the sample index $t_n \triangleq t + P - M + n$ and $w(t_n)$ is the windowing function.

As mentioned in the Introduction, existing HD methods aim to detect howling by first identifying magnitude peaks in the microphone signal spectrum and

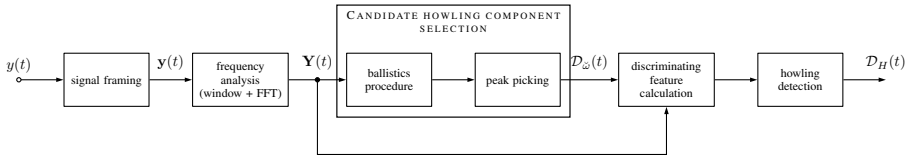


Figure 8.4: HD solution scheme: in the proposed approach, the candidate howling component selection is omitted.

then distinguishing howling from desired speech or music components only for this subset $\mathcal{D}_{\breve{\omega}}(t)$ of candidate howling frequencies. The candidate selection procedure, consisting of a magnitude spectrum peak-picking step that is sometimes preceded by a so-called ballistics procedure [96], is indicated by the large box in Fig. 8.4. Afterwards, HD is achieved by comparing a howling-specific spectral and/or temporal feature – or some logical combination of features – to a prespecified detection threshold. The HD block then outputs a set of notch filter design parameters $\mathcal{D}_H(t)$, *e.g.*, including the desired notch filter center frequency, bandwidth, and notch depth [15].

In this paper, we propose a modification of the common HD solution scheme in which we remove the candidate howling selection and directly calculate the HD features on the STFT coefficients of the microphone signal. In other words, the microphone signal STFT vector $\mathbf{Y}(t)$ is forwarded to the discriminating feature calculation block only via the lower arrow in Fig. 8.4. This can also be interpreted as assigning all STFT frequency bins to be candidate howling frequencies, effectively broadening the capacity of the HD block to include early-howling and ringing detection.

8.2.3 State-of-the-art howling detection (HD) features

Here we revisit the most commonly used HD features summarized in [15], as these will serve as a baseline for evaluating the different contributions of this paper. From a spectral point of view, howling is of a purely sinusoidal nature. As shown in Fig. 8.2 in the STFT domain, howling can be identified as a narrowband, persisting frequency component exhibiting high power with respect to its neighboring frequency components. Another property discriminating howling from speech and music components is the fact that a howling components does not have harmonics, unless loudspeaker saturation and therefore clipping is reached.

These properties gave birth to four spectral HD features each comparing the power of the ith STFT time-frequency bin to a specific reference power:

Peak-to-Threshold Power Ratio (PTPR) [96–99]: the reference power is a fixed absolute power threshold P_0. This feature hence assumes that desired speech and music components are power-limited.

$$\mathrm{PTPR}(\omega_i, t) = 10 \log_{10} \frac{|Y(\omega_i, t)|^2}{M P_0}. \tag{8.8}$$

Peak-to-Average Power Ratio (PAPR) [96–108]: the reference power is the average microphone signal power $\hat{P}_y(t)$. This feature relaxes the power-limited assumption of the PTPR feature by allowing any power for the desired speech and music components, but still considers howling only when it has a large power relative to the desired signal components, which excludes it to be used for early-howling and ringing detection.

$$\mathrm{PAPR}(\omega_i, t) = 10 \log_{10} \frac{|Y(\omega_i, t)|^2}{\hat{P}_y(t)} \tag{8.9}$$

with

$$\hat{P}_y(t) = \frac{1}{M} \sum_{k=0}^{M-1} |Y(\omega_k, t)|^2 . \tag{8.10}$$

Peak-to-Neighboring Power Ratio (PNPR) [109–112]: the reference power is the power of the mth neighboring frequency component. This feature exploits the sinusoidal nature of howling, and the value for m is chosen depending on the STFT resolution and accuracy, which in turn depend on the STFT size M and the applied windowing.

$$\mathrm{PNPR}(\omega_i, t, m) = 10 \log_{10} \frac{|Y(\omega_i, t)|^2}{|Y(\omega_i + 2\pi m/M, t)|^2}. \tag{8.11}$$

Peak-to-Harmonic Power Ratio (PHPR) [97, 98, 113, 114]: the reference power is the power of the mth (sub)harmonic. This feature exploits the fact that howling does not have harmonic components, in the absence of clipping saturation.

$$\mathrm{PHPR}(\omega_i, t, m) = 10 \log_{10} \frac{|Y(\omega_i, t)|^2}{|Y(m\omega_i, t)|^2}. \tag{8.12}$$

In addition to these spectral features, two temporal HD features have been proposed in literature:

Interframe Peak Magnitude Persistence (IPMP) [102–104,113–116]: this feature captures howling persistence over time around a certain frequency. It does that by counting the number of frames out of the past $\mathcal{Q}_M$ frames in which the ith frequency bin is in the set $\mathcal{C}_\omega(t)$ that keeps track of the C largest-magnitude frequency bins at each time t.

$$\mathrm{IPMP}(\omega_i, t) = \frac{\sum_{j=0}^{\mathcal{Q}_M} [\omega_i \in \mathcal{C}_\omega(t - jP)]}{\mathcal{Q}_M}. \tag{8.13}$$

Interframe Magnitude Slope Deviation (IMSD) [110–112]: this feature represents the frame-wise variation of the slope of the logarithmic magnitude increase of a frequency component as a function of time, which is expected to be nearly constant for a howling component while being time-varying for speech and music components. It is calculated as the log-spectral magnitude (dB-scale) slope deviation over the past $\mathcal{Q}_M$ frames for the ith frequency bin,

$$\begin{aligned}\mathrm{IMSD}(\omega_i, t) = \frac{1}{\mathcal{Q}_M - 1} \sum_{m=1}^{\mathcal{Q}_M - 1} \Bigg[\\ \frac{1}{\mathcal{Q}_M} \sum_{j=0}^{\mathcal{Q}_M - 1} \frac{1}{\mathcal{Q}_M - j} \times 20 \log_{10} \frac{|Y(\omega_i, t - jP)|}{|Y(\omega_i, t - \mathcal{Q}_M P)|} \\ - \frac{1}{m} \sum_{j=0}^{m-1} \frac{1}{m - j} \times 20 \log_{10} \frac{|Y(\omega_i, t - jP)|}{|Y(\omega_i, t - mP)|} \Bigg].\end{aligned} \tag{8.14}$$

It is averaging the differences between long-term and short-term weighted slope averages corresponding to the second and the third lines of (8.14). The IMSD value is expected to approach zero for howling components, representing a linear slope increase on a log-magnitude scale. Although it was found to perform well on the example studied in [15], it is a feature that is extremely sensitive to the detection threshold choice. This can be noticed by investigating the thresholds used for plotting its ROC curve in [15], and will also be illustrated in the simulation results in Section 8.6.

It is noticeable that indeed almost all of these features contain the word *peak* in their names. This is due to the fact that these features were developed to detect howling only for a few candidate magnitude spectrum peaks. For the reasons introduced earlier, in this work, the features are calculated for all frequency bins which is already reflected in the above feature definitions. Out of the features presented above, only the IPMP is a normalized feature, *i.e.*, yielding values

between zero and one independently from the signal and experiment parameters. This is a desirable property as it facilitates the choice of a detection threshold. Therefore, a feature normalization will be carried out for the other features, as will be explained in more detail in Section 8.6.

Apart from the above features that are widely used in HD, a few other features have been proposed more recently. In [91], a spectral flatness measure (SFM) is used for identifying frequency bands where the magnitude spectrum is peaky, which is considered as an indication of howling. This feature is however hard to compare to the HD features used in NHS, as the method in [91] is an AEQ rather than a NHS method, since the detection and gain reduction is performed in wider frequency bands than those typically used in NHS. Another feature representing spectral peakiness based on the Generalised Teager-Kaiser Operator (GTKO) was proposed in [38]. When combined with the PHPR to reduce false alarms, the GTKO feature was shown to outperform other features in a HA application. In [40], the features discussed in [15] were combined with a Voice Activity Detection (VAD), again to reduce false alarms. A machine learning approach to HD based on Support Vector Machines (SVMs) was proposed in [48], but only for taking a binary decision on whether howling occurs or not, without estimating the howling frequency. Finally, some papers have evaluated the combination of two or more features through a logical conjunction operation. The rationale behind the use of multiple features is that the detection threshold can be lowered, aiming to detect more howling instances without increasing the number of false alarms. A strategy for combining multiple features based on the single-feature HD performance was proposed in [15], and combinations between the standalone features PHPR, PNPR and IMSD were evaluated. Instead of a logical conjuction, features can also be combined by linear combination, *e.g.*, the feedback existence probability (FEP) feature [110–112] which is a linear combination of a PNPR-related feature and an IMSD-related feature.

8.3 Proposed howling detection feature

A novel feature for HD is proposed here, perhaps somewhat surprisingly inspired by a feature proposed earlier for the problem of musical note onset detection [33]. At first sight, the detection of howling and musical note onsets may seem unrelated. However, in both problems, one is looking for lines in the spectrogram. Musical note onsets are characterized by a short and broadband increase in signal energy and thus become visible as vertical lines in the spectrogram [33]. Howling, on the other hand, manifests itself as a persisting narrowband signal component observed as a horizontal line in the spectrogram, see *e.g.*, Fig. 8.2. This analogy has motivated us to design a HD feature based on the same

principles used in the development of the so-called NINOS^2 feature proposed in [33] for note onset detection.

The key idea behind the NINOS^2 feature is spectral sparsity: a vector containing STFT magnitude coefficients for a single time frame will have few large elements during the tonal part of a note while it has elements of equal magnitude during a note onset. This implies that such vector is less sparse for an onset time frame than for a non-onset time frame, hence an inverse sparsity measure (yielding a larger value for a less sparse vector) is a suitable feature to detect note onsets. Indeed, according to [78], two basic conditions that should be satisfied by any sparsity measure are the following:

1. The most sparse vector is the one having all its energy (magnitude) concentrated in one coefficient.
2. The least sparse vector is the one where all coefficients have the same magnitude.

It is argued in [33] that an inverse sparsity measure can be obtained by considering the ratio of two vector norms as follows, for an arbitrary length-M vector $\mathbf{x} = [x_0, \ldots, x_{M-1}]$,

$$\mathcal{S} = \frac{\|\mathbf{x}\|_p}{\|\mathbf{x}\|_q} = \frac{\left(\sum_{m=0}^{M-1} |x_m|^p\right)^{\frac{1}{p}}}{\left(\sum_{m=0}^{M-1} |x_m|^q\right)^{\frac{1}{q}}}, \tag{8.15}$$

where $p < q$. For note onset detection, this inverse sparsity measure is combined with an energy measure since note onsets often also exhibit a short-term energy increase, resulting in a feature coined as identifying note onsets based on spectral sparsity (INOS^2) [33],

$$\mathcal{I} = \|\mathbf{x}\|_2 \cdot \frac{\|\mathbf{x}\|_p}{\|\mathbf{x}\|_q}. \tag{8.16}$$

With the choice $p = 1, q = 2$, the INOS^2 feature reduces to the ℓ_1-norm $\|\mathbf{x}\|_1$, which is indeed a widely used joint energy and inverse sparsity measure. In [33], the norms are chosen as $p = 2, q = 4$, and the INOS^2 feature is normalized to yield a value $\in [0, 1]$, resulting in the NINOS^2 feature.

In the context of HD, one could similarly use a joint energy and inverse sparsity measure, but now considering a vector representing the time variation (over

$\mathcal{Q}_M$ frames) of the STFT in a single frequency bin, *i.e.*,

$$\mathbf{Y}_T(\omega_i, t) = \begin{bmatrix} Y(\omega_i, t - \mathcal{Q}_M + 1) & \dots & Y(\omega_i, t) \end{bmatrix}^T. \tag{8.17}$$

This vector should not be confused with the vector $\mathbf{Y}(t)$ defined in (8.7): considering the microphone signal STFT over M frequency bins and $\mathcal{Q}_M$ time frames, as a $M \times \mathcal{Q}_M$ matrix $\mathbf{Y}$, then $\mathbf{Y}(t)$ defined in (8.7) represents a column of that matrix while $\mathbf{Y}_T(\omega_i, t)$ defined in (8.17) represents a row of that matrix. The use of a joint energy and inverse sparsity measure for HD would express the fact that howling is a high-energy component and that it persists over a relatively long period in time. The high-energy property is however only discriminative for howling components that have already built up a significant energy and can thus be expected to be clearly audible. This goes against our objective of developing a HD feature that is capable of detecting early howling and ringing. Therefore we explicitly choose to remove the energy measure and only retain the inverse sparsity measure in the proposed HD feature. Using $p = 2, q = 4$ as in [33], the inverse sparsity measure in (8.18) applied to the vector $\mathbf{Y}_T(\omega_i, t)$ defined in (8.17) yields the following HD feature,

$$\mathcal{S} = \frac{\|\mathbf{Y}_T(\omega_i, t)\|_2}{\|\mathbf{Y}_T(\omega_i, t)\|_4} = \frac{\left(\sum_{m=0}^{\mathcal{Q}_M - 1} |Y(\omega_i, t - m)|^2 \right)^{\frac{1}{2}}}{\left(\sum_{m=0}^{\mathcal{Q}_M - 1} |Y(\omega_i, t - m)|^4 \right)^{\frac{1}{4}}}. \tag{8.18}$$

For an easier choice of the detection threshold, the above measure $\mathcal{S}$ is normalized to have a value $\in [0, 1]$ ranging from the most sparse (0) to the least sparse (1) vector. This can be achieved by defining the minimum and maximum values that $\mathcal{S}$ can take and then computing a normalized measure as follows,

$$\mathcal{N} = \frac{\mathcal{S} - \mathcal{S}_{\min}}{\mathcal{S}_{\max} - \mathcal{S}_{\min}}. \tag{8.19}$$

The values $\mathcal{S}_{\min}$ and $\mathcal{S}_{\max}$ can be found by considering two extreme cases of an arbitrarily scaled length-$\mathcal{Q}_M$ vector: the sparsest possible vector $[a, 0, 0, \dots, 0]$ having $\mathcal{S} = 1 \triangleq \mathcal{S}_{\min}$ and the least sparse vector $[a, a, a, \dots, a]$ having $\mathcal{S} = \sqrt[4]{\mathcal{Q}_M} \triangleq \mathcal{S}_{\max}$. By substituting these values in (8.19) we obtain the proposed NINOS^2-T feature

$$\mathcal{N} = \frac{1}{\sqrt[4]{\mathcal{Q}_M} - 1} \left(\frac{\|\mathbf{Y}_T(\omega_i, t)\|_2}{\|\mathbf{Y}_T(\omega_i, t)\|_4} - 1 \right). \tag{8.20}$$

Analogous to the onset detection function (ODF) defined in musical note onset detection, which is a highly sub-sampled, real and nonnegative version of the

original signal exhibiting amplitude peaks at the note onsets, here we introduce a howling detection function (HDF) that can be compared to a detection threshold to achieve HD. In contrast to the ODF which is a function of time only, the HDF is a function of time and frequency that can be thought of as a sub-sampled, real and nonnegative version of the microphone signal STFT, exhibiting amplitude peaks in time-frequency bins where howling is found to occur. The NINOS^2-T HDF $\mathcal{N}(\omega_i, t)$ is simply equal to the NINOS^2-T feature in (8.21), considered as an explicit function of time t and frequency ω_i,

$$\mathcal{N}(\omega_i, t) = \frac{1}{\sqrt[4]{\mathcal{Q}_M} - 1} \left(\frac{\|\mathbf{Y}_T(\omega_i, t)\|_2}{\|\mathbf{Y}_T(\omega_i, t)\|_4} - 1 \right). \tag{8.21}$$

8.4 Howling detection dataset

In order to test the proposed HD feature and compare it with different existing features, we created a large and diverse database containing a variety of speech and music excerpts corrupted by simulated howling. The database creation is detailed in Chapter 7 and here we summarize only a few of its parameters. The database contains in total 28 music files and 30 speech files. Each file is a 20 s excerpt where the howling is simulated to start between the 8th and 9th second. This is simulated by feeding the music or speech source signal to a closed-loop system with a varying broadband gain as depicted in Fig. 7.2. A total of 8 acoustic impulse responses (AIRs) were used for the simulations, hence covering a wide range of howling frequencies. The final dataset[1] contains fewer excerpts than the number of excerpts originally generated, as it went through a pruning step conducted by three experts, aiming to eliminate unsuitable examples (exhibiting howling at multiple howling frequencies or no howling at all).

8.5 Performance evaluation procedure

The aim of HD is to identify frequencies at which howling occurs and to estimate the time interval during which it occurs. There are several ways in which the performance of a HD method can be quantified. A simple performance evaluation procedure can be achieved by considering the HD problem as an event detection problem where, in each time-frequency bin of the microphone signal STFT $Y(\omega_i, t)$, the howling event may or may not occur. In this way, the

[1] available for download at `ftp://ftp.esat.kuleuven.be/stadius/mshehata/datasets/Howling_Corrupted_Music_Speech_HCMS.zip`

HD problem becomes a binary classification problem and the performance of an HD method can be carried out in a binary classification framework. A binary annotation (howling / no howling) is then needed for each time-frequency of the microphone signal STFT. Due to the fact that our dataset is created by simulating howling in a closed-loop system with a known feedback path transfer function $F(q)$ (relating to the AIR used in a specific howling simulation) and a known forward path transfer function $G(q)$ (relating to the specified gain profile), the binary annotation can be automatically generated by making use of the Nyquist stability criterion introduced in (8.2)-(8.3). Note that this annotation depends on the selected AIR and gain profile, as well as on the STFT frame size and hop size, but does not depend on the selected source signal as the Nyquist stability criterion is source-signal-independent.

In previous work, *e.g.*, [15], the evaluation of HD features was compared on a set of candidate howling frequencies using the ROC curve and the ROC-AUC measure. In addition to the limitation of the use of a candidate set, as already explained in Section 8.1, also the use of the ROC curve has its shortcomings when aiming to visually compare the performance of the different features, particularly when considering all STFT frequency bins as candidate howling frequencies. This happens because the ROC curve has been designed for the evaluation of binary classification methods operating on datasets with balanced classes (*i.e.*, both types of events, howling / no howling, occur with equal probability). As this is generally not the case, *i.e.*, howling occurs much less frequently than no howling, we will instead use the PR curve and the PR-AUC measure to evaluate the different features on the entire STFT frequency grid, and provide a comparison with the ROC-based metrics. We will also include the best F_1-score (F_1) corresponding to the optimal detection threshold $\theta^\star$. For more details about the PR-based metrics and their advantages over the traditionally used ROC-based metrics, please refer to Section 2.5.

8.6 Simulation results

In this section, we will compare the results of the proposed NINOS^2-T feature to the six baseline features introduced in Section 8.2, *i.e.*, the four spectral features (PTPR, PAPR, PNPR and PHPR) and the two temporal features (IPMP and IMSD). In the performance comparison, we will also include a variant of the NINOS^2-T feature that does include an energy measure. This variant will be referred to as the NINOS^2 feature as it is defined precisely like the NINOS^2 feature proposed for note onset detection in [33], but operating on rows rather than columns of the STFT matrix $\mathbf{Y}$.

Before discussing the simulation results, we first summarize the different feature parameters and simulation details. The microphone signal is divided into overlapping frames with a hop size equivalent to a frame rate of 50 frames-per-second (fps). Four different STFT frame sizes $M = 512$, 1024, 2048 and 4096 samples, were tried out and the optimal value was selected for each feature separately. All features except IPMP are calculated on the log-magnitude of a single sideband of the spectrum. In case of the IPMP, it is not necessary to use the log-magnitude as it considers magnitude spectrum maxima which remain the same for linear or logarithmic magnitude. With IPMP, we kept the number of tracked spectral peaks C equal to 3 according to what is suggested in [15]. For PNPR and PHPR we limited our experiment to the three best scoring sets of neighbors ($m = \pm\{2\}$, $\{2, 3\}$ and $\{2, 3, 4\}$) and harmonics ($m = \{2\}$, $\{2, 3\}$ and $\{2, 3, 4\}$) in [15]. Each one of the temporal features, including NINOS^2-T and NINOS^2, was run with the number of preceding frames $\mathcal{Q}_M = 4$, 8, 16, 32, 64 and 96. As previously stated, only the IPMP and NINOS^2-T features yield normalized values between 0 and 1. For the remaining features the resulting feature spectrum, *i.e.*, the vector of feature values for all frequency bins in a single time frame, has to be normalized before applying the thresholding, to avoid that optimal detection threshold values would have to be selected for each test signal separately. The normalization of the feature spectrum requires signal-dependent normalization parameters, *i.e.*, maximum and minimum spectral magnitudes, which are here calculated from the first 2 seconds of each test signal. As a consequence, the first 2 seconds of each test signal are excluded from the evaluation. Moreover, and to allow for a fair comparison, the first few seconds corresponding to the largest value for $\mathcal{Q}_M$, *i.e.*, $\mathcal{Q}_M = 96$ frames, should also be excluded from the evaluation for the temporal features. These exclusions will hardly affect the resulting HD performance measures as the howling in our dataset starts occurring only as from the 8th to 9th second of the test signal.

For each test signal a binary annotation is automatically generated by using knowledge of the howling frequency $\tilde{\omega}_h$, the gain start time T_s (= 8 s in our dataset) and the gain rise time T_r (= 1 s in our dataset). The howling frequency $\tilde{\omega}_h$ is calculated from the Nyquist stability criterion given the selected AIR and setting the forward path gain equal to the MSG. As the howling frequency $\tilde{\omega}_h$ will generally not lie on the STFT frequency grid $\omega_k = (2\pi k)/M$, $k = 0, \ldots, M-1$, the two frequency bins on either side of the howling frequency $\tilde{\omega}_h$, starting from time $(T_s + 0.5T_r)$ to the end of the test signal, are annotated as howling events. The binary annotation $a(\omega_i, t)$ is thus defined as follows,

$$a\left(\omega_i, t\right) = \begin{cases} 1, & \text{if} \begin{cases} t \geq T_s + \frac{T_r}{2} \\ \left\lfloor \frac{M\tilde{\omega}_h}{2\pi} \right\rfloor \leq i \leq \left\lceil \frac{M\tilde{\omega}_h}{2\pi} \right\rceil \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{8.22}$$

with $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denoting the floor and ceiling functions, respectively. Since every howling event is generally represented by two frequency bins, we count one true positive if howling is detected in one or both of these bins. Also, in case two neighboring bins are erroneously detected to contain howling, only one false positive is counted. To generate the PR curves and their corresponding PR-AUC measures, a grid of thresholds $\theta \in [0, 1]$ with a grid resolution of 0.05 is used. Added to this, two additional thresholds $\theta = \pm\infty$ are used to make sure the curves include the extreme cases. All presented curves are interpolated to produce $I = 50$ points in addition to the calculated ones. The interpolated points are computed for equally spaced true positive (TP) values $\in [0, GT_+]$, with GT_+ being the number of ground truth positives. For each of the compared features, the thresholding is applied in two different ways. The first way is to directly compare all time-frequency feature values to the detection threshold. The second way is to first select the S_c largest values of each feature spectrum per time frame and only comparing these to the detection threshold. This peak picking should not be confused with the STFT magnitude peak picking that is used in state-of-the-art HD methods, see Fig. 8.4, as it occurs *after* and not before the feature calculation. When $S_c = M$, the second thresholding approach becomes equivalent to the first one, *i.e.*, all feature values are compared to the detection threshold. In the sequel, the following values will be considered for all features under comparison: $S_c \in \{1, 3, M\}$.

Each of the features depends on a few parameters for which numerical values need to be chosen: the STFT frame size M, the number of feature values S_c used per frame, the number of past frames Q_M for the temporal features, and the neighbor and harmonics indices m for the PNPR and PHPR features, respectively. The choice of these parameters is determined here using a 5-fold cross-validation setup applied separately on the music and speech datasets. The excerpts in each of these two datasets are organized into 5 folds and 5 simulation rounds are executed where in each round, 4 folds are used for tuning the feature parameters while the 5th fold is used for validation. In each round and for each feature, the parameters resulting in the highest PR-AUC value for the tuning folds, are used to quantify the feature performance on the validation fold. In this way, for each feature, 5 PR-AUC values are obtained (one value for each validation fold), the statistics of which are summarized in the box plots in Figure 8.5. In all the following analyses, two evaluation scenarios are considered: full HD and early HD evaluation. While in the full HD evaluation, the performance measures are calculated for the full excerpt (even though excluding the first few seconds as motivated earlier), in the early HD evaluation, the performance measures are calculated only up to 5 s after the theoretical start of the howling, *i.e.*, up to time $8.5 + 5 = 13.5$ s instead of up to 20 s as in the full evaluation. In this way, the early HD evaluation scenario provides an indication of the capability of the different features to detect early

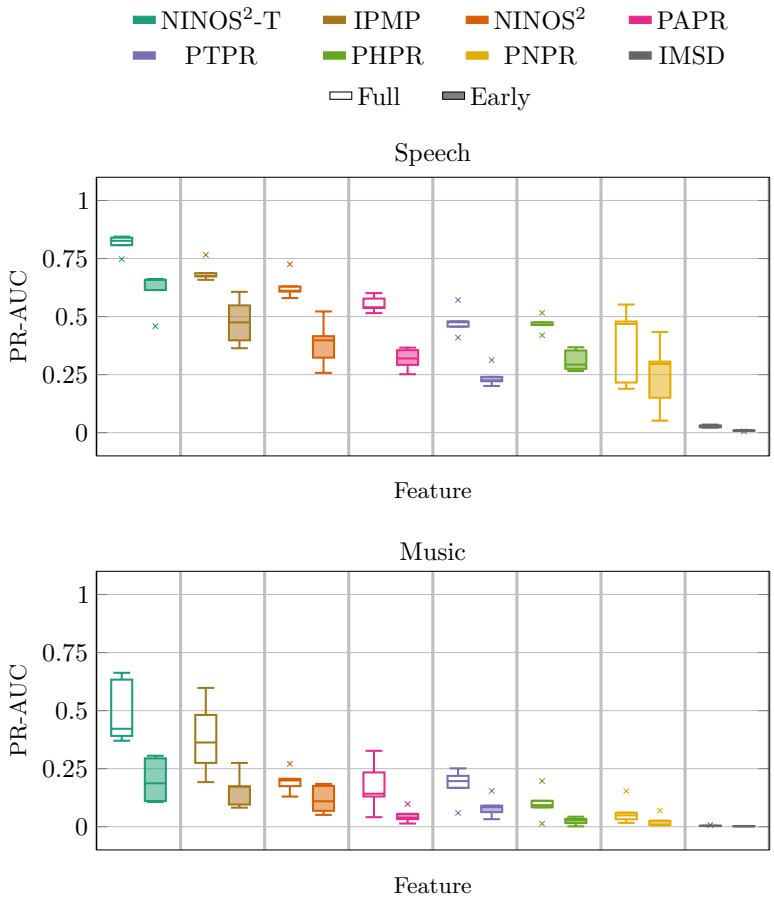


Figure 8.5: Cross-validation PR-AUC results for full and early HD evaluation of the different features when applied on the speech (top) and music (bottom) datasets.

howling and ringing.

The NINOS^2-T feature consistently yields the highest average PR-AUC value for both speech and music datasets and for both full and early HD evaluation. While the performance ranking of the three best features, *i.e.*, the NINOS^2-T, IPMP and NINOS^2 features, remains the same over all datasets and evaluation scenarios, there is a minor variability in the performance ranking of the other features. The IMSD feature consistently shows the worst performance, which is presumably due to its high sensitivity to the detection threshold choice. Overall,

Table 8.1: Optimal parameter values per feature (speech dataset).

Feature	M	$\mathcal{Q}_M$	S_c	m
NINOS^2-T	1024	32	1	–
NINOS^2	1024	16	1	–
PTPR	512	–	1	–
PAPR	2048	–	M	–
PHPR	2048	–	3	$\{2, 3, 4\}$
PNPR	2048	–	M	$\pm\{2, 3, 4\}$
IPMP	2048	64	M	–
IMSD	512	8	3	–

it is clear how the HD problem is more challenging for music than for speech, in particular for the detection of early howling.

The optimal parameter values for each feature resulting from the cross-validation procedure are listed in Table 8.1 for the speech dataset and in Table 8.2 for the music dataset. These optimal parameter values are the values resulting in the highest average PR-AUC value considering the early HD evaluation over all 5 validation folds. It can be observed that the proposed NINOS^2-T feature is the only feature for which it is beneficial to set $S_c = 1$ in all evaluation scenarios, indicating that its highest feature value across all frequency bins in a single time frame consistently seems to point to the most probable occurrence of howling, which is a beneficial property for a HD feature.

In Table 8.3 and Table 8.4 we present PR-AUC and ROC-AUC values as well as the best achievable F_1-score for each of the features, using the optimal parameter values for each feature. The NINOS^2-T feature is the only feature crossing the 50 % PR-AUC value in the full HD evaluation for both speech and music. For early HD evaluation, the NINOS^2-T feature only crosses the 50 % PR-AUC value for speech but not for music, where all features can be observed to perform poorly. We also observe a strong correlation between the PR-AUC measure and the F_1-score which confirms that the PR evaluation framework is more suitable than the ROC evaluation framework in the HD context. Only for the IPMP and NINOS^2 features in the speech dataset, the performance ranking based on the PR-AUC measure is different from the ranking by the F_1-score.

A deeper insight into the values reported in Table 8.3 and Table 8.4 can be

Table 8.2: Optimal parameter values per feature (music dataset).

Feature	M	$\mathcal{Q}_M$	S_c	m
NINOS^2-T	1024	96	1	–
NINOS^2	512	16	M	–
PTPR	512	–	M	–
PAPR	512	–	M	–
PHPR	512	–	M	$\{2, 3, 4\}$
PNPR	2048	–	M	$\pm\{2, 3\}$
IPMP	4096	64	1	–
IMSD	512	16	3	–

obtained by visually inspecting the associated ROC and PR curves, which are shown in Fig. 8.6 and Fig. 8.7 for the speech and music dataset, respectively. For an easier interpolation of the PR curves, the evaluation metrics are calculated in a micro average setup, *i.e.*, TPs and false positives (FPs) are aggregated for the whole dataset then used for computing the other metrics and visualizations. Alternatively, the interpolated PR curves for the different test files can be macro-averaged using several techniques [59], but this is out of the scope of this work. In contrast to the PR curves, the ROC curves fail to differentiate the performance of the different features. Due to the huge class imbalance, the only practical detection thresholds would be the ones resulting in points lying on the vertical true positives rate (TPR) axis of the ROC curve, as a small displacement away from this axis immediately results in a very high false positives rate (FPR) value. As such, the visual comparison of the ROC curves is reduced to finding the highest point on the curve nearby the vertical axis, which is not always feasible without zooming in. The reason why the baseline – marking the random guessing performance – for the PR curves is constantly omitted, is because it is a constant line almost equal to zero - in the order of 10^{-3}. The points in the ROC and PR curves originating from the actual performance evaluation are indicated with markers to differentiate them from the points obtained by interpolation. Focusing on the three best performing features in Fig. 8.6 and Fig. 8.7, *i.e.*, NINOS^2-T, NINOS^2, and IPMP, it can be observed that the marked points in the PR curve are spread further apart for the IPMP feature compared to the NINOS^2-T and NINOS^2 features, which indicates that the latter features are more robust to detection threshold variations. Another interesting observation is that the NINOS^2-T and NINOS^2 features tend to

yield PR curves that are more closely approaching the top-right corner of the PR plane compared to the IPMP feature. This indicates that the NINOS^2-T and NINOS^2 features are capable of providing a better compromise between precision (P) and recall (R), despite perhaps showing a lower PR-AUC value than the IPMP feature in some scenarios.

Table 8.3: Speech dataset evaluation using the best parameterisation per feature. It compares the best F_1, PR-AUC and ROC-AUC for the full and early evaluations.

	Full			Early		
Feature	Best F_1	PR-AUC	ROC-AUC	Best F_1	PR-AUC	ROC-AUC
NINOS^2-T	0.88	0.82	0.93	0.74	0.63	0.86
NINOS^2	0.77	0.65	0.90	0.58	0.40	0.82
PTPR	0.60	0.49	0.83	0.41	0.25	0.76
PAPR	0.63	0.56	0.99	0.45	0.31	0.98
PHPR	0.59	0.47	0.76	0.46	0.31	0.68
PNPR	0.55	0.44	0.94	0.41	0.27	0.88
IPMP	0.68	0.69	0.95	0.50	0.47	0.91
IMSD	0.13	0.03	0.66	0.07	0.01	0.60

Table 8.4: Music dataset evaluation using the best parameterisation per feature. It compares the best F_1, PR-AUC and ROC-AUC for the full and early evaluations.

	Full			Early		
Feature	Best F_1	PR-AUC	ROC-AUC	Best F_1	PR-AUC	ROC-AUC
NINOS^2-T	0.70	0.53	0.83	0.42	0.21	0.69
NINOS^2	0.36	0.19	0.95	0.27	0.13	0.94
PTPR	0.33	0.17	0.96	0.19	0.08	0.95
PAPR	0.33	0.16	0.96	0.17	0.05	0.94
PHPR	0.18	0.07	0.95	0.09	0.03	0.93
PNPR	0.16	0.05	0.87	0.08	0.02	0.80
IPMP	0.47	0.36	0.71	0.23	0.15	0.61
IMSD	0.05	0.01	0.56	0.03	0.00	0.54

In Fig. 8.8, the box-plots show how the feature performance in terms of the of PR-AUC is distributed for the different test signals. Again, we can see that the NINOS^2-T feature yields the best average and the best worst-case performance for both datasets (speech / music) and HD evaluation scenarios (full / early). It also shows the least variance over the different test signals. For the challenging

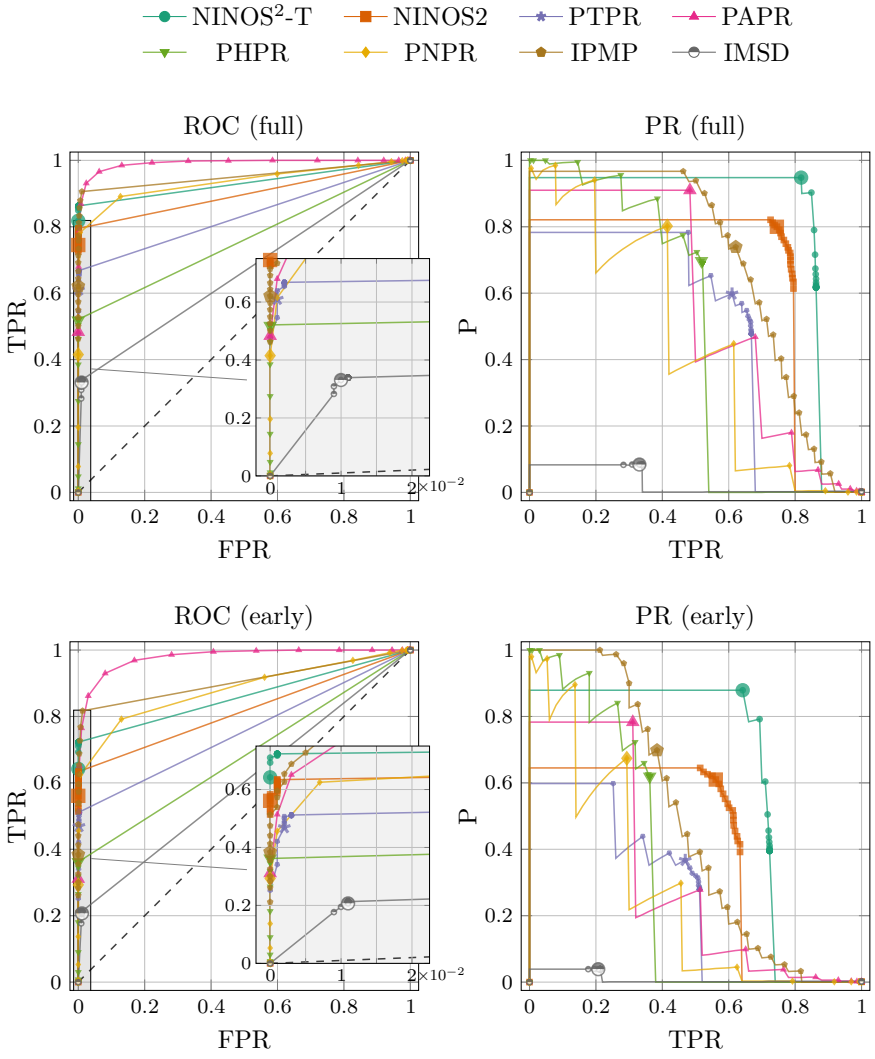


Figure 8.6: ROC and PR curves for the speech dataset.

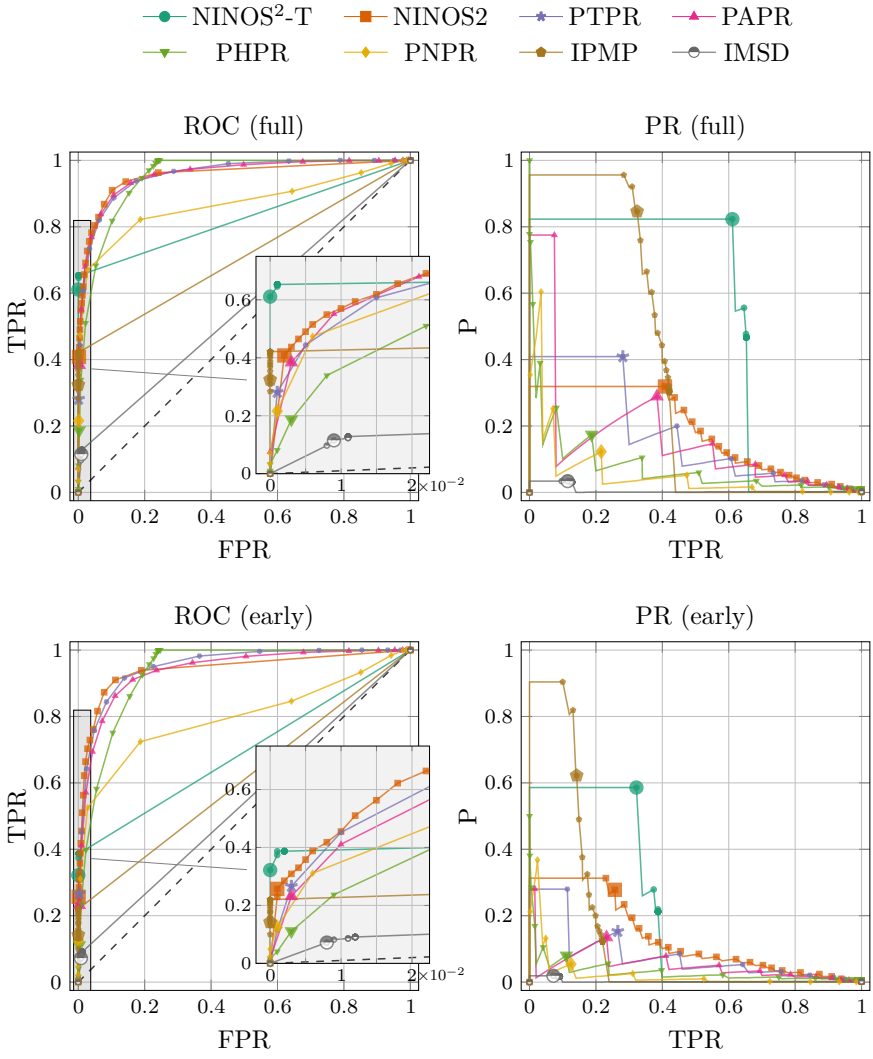


Figure 8.7: ROC and PR curves for the music dataset.

problem of HD in music signals, all features except for the best three features perform on average worse than the PTPR feature, which represents a simple thresholding of the STFT magnitude spectrum.

Finally, Fig. 8.9 shows a spectrogram-type snippet of the HDFs for two challenging test examples. The leftmost plots show HDFs for a speech example containing a ringing artefact around 800 Hz, having a lower power than the dominating speech components. The rightmost plots show a music example containing many desired tonal components in addition to a low-power howling component at 260 Hz. Only the NINOS^2-T feature was able to capture this howling very close to its starting point, between the 8th and the 9th second. Overall, the difference between the spectral and temporal features can be clearly observed, since the spectral HDFs have a more "vertical" and granular structure whereas the temporal HDFs have a more "horizontal" and smoother structure.

8.7 Conclusions & Future work

In this work we have addressed three challenges relating to the HD problem: feature design, dataset design, and performance evaluation. A novel HD feature based on spectral sparsity, NINOS^2-T, was proposed and shown to yield better average and better worst-case performance than state-of-the-art features across various evaluation scenarios involving music and speech signals and considering also detection of early howling and ringing. The NINOS^2-T feature is a normalized feature, facilitating the choice of a suitable detection threshold in the range $[0, 1]$, and was experimentally shown to be more robust to threshold variability than existing features. Simulations were carried out on a new, automatically annotated HD dataset which is larger and more diverse than existing datasets. Finally, a novel evaluation procedure is put forward in which all time-frequency bins are considered as candidate howling components, to overcome the limitation of the traditional candidate-based approach to HD which is not suitable for the detection of early howling and ringing artefacts. The high class imbalance inherent in the HD problem makes the traditional ROC-based evaluation unsuitable. Therefore, a PR-based evaluation was proposed which facilitates the performance comparison of different features independently of the detection threshold.

Future work includes a noise-robustness assessment and improvement of the proposed NINOS^2-T feature and its combination with other features in a multiple-feature HD approach particularly targeting the challenging problem of HD in music applications. Finally, the new HD dataset is expected to be suitable also for the development and evaluation of data-driven HD methods.

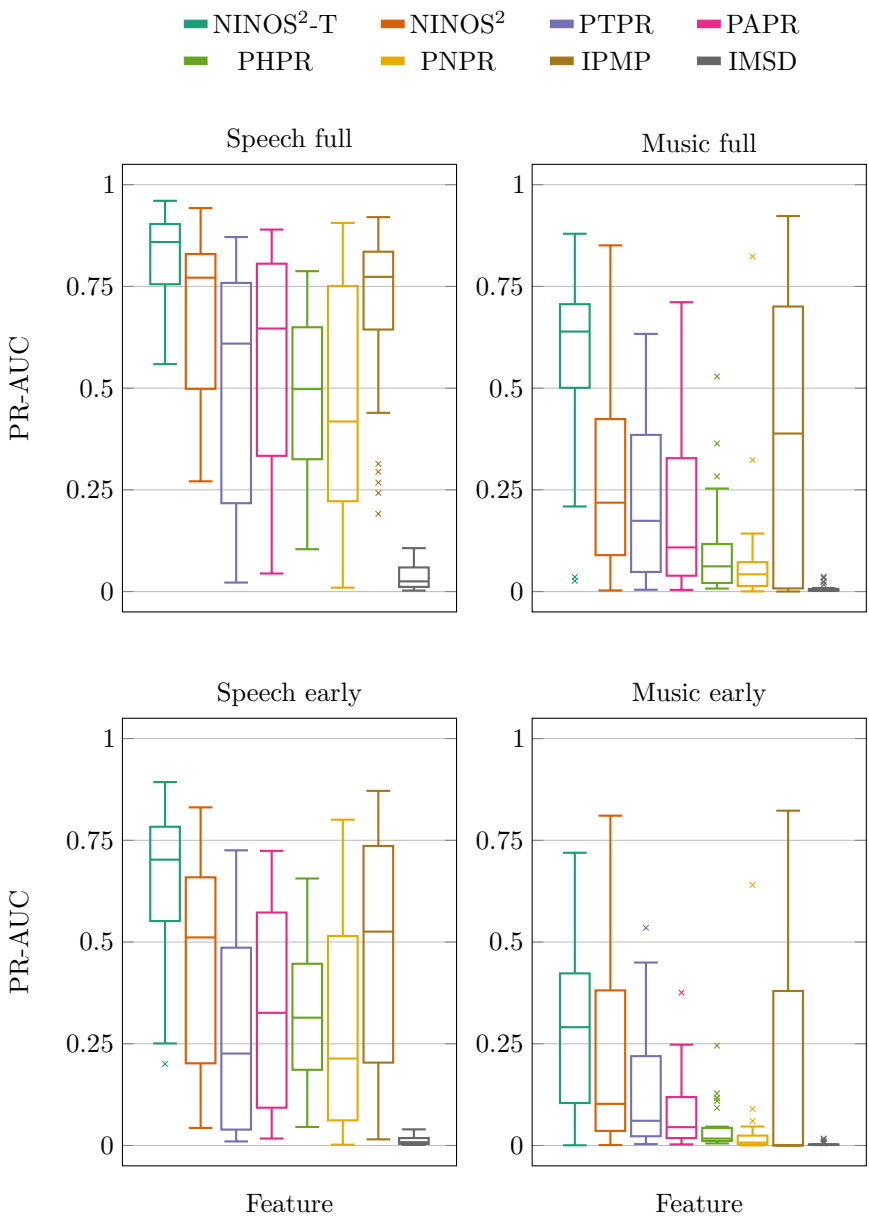


Figure 8.8: Comparison of PR-AUC statistics for the different test examples per dataset (speech / music) and HD evaluation scenario (full / early).

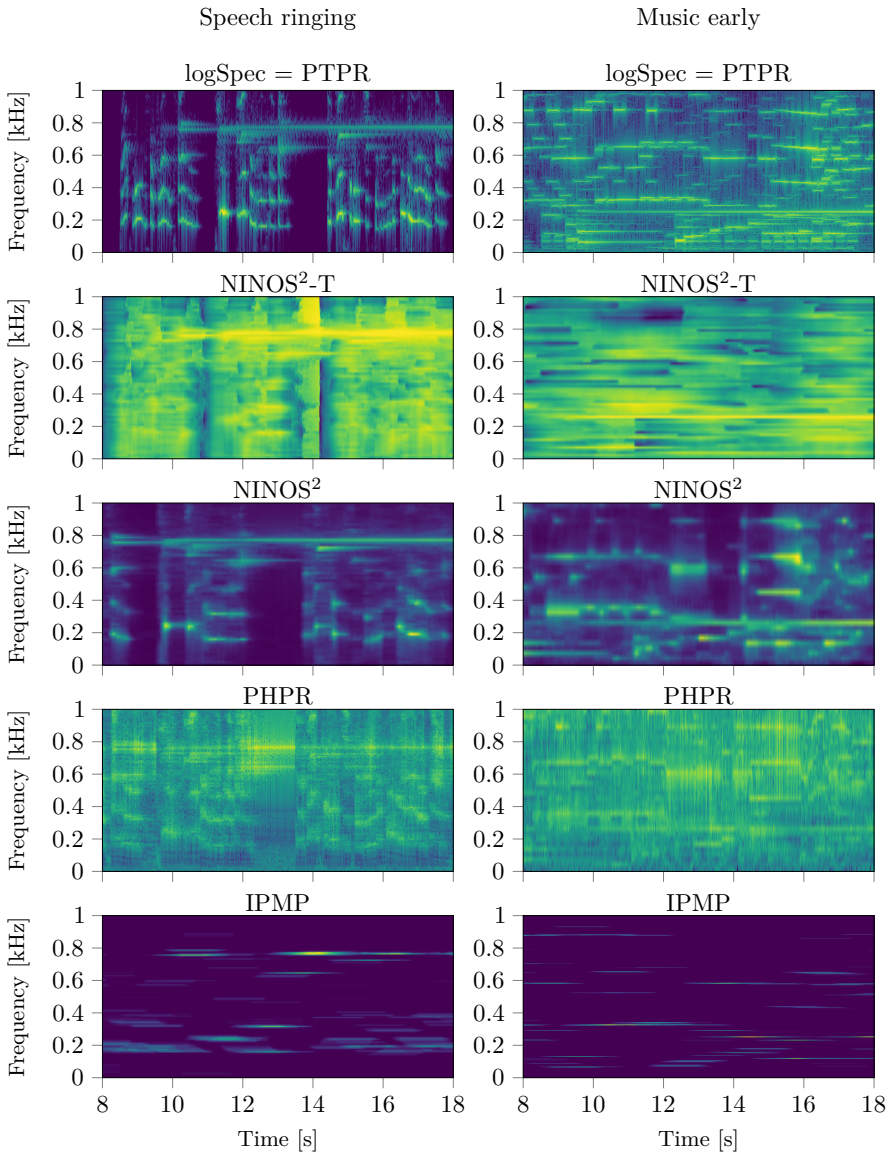


Figure 8.9: Comparison of the HDFs for (left) a speech example exhibiting a ringing artifact around 800 Hz and (right) a music excerpt exhibiting a low power howling around 260 Hz.

Part IV

Conclusion

Chapter 9

Conclusion and future work

In this chapter, we first discuss the different contributions of the thesis. Then, we will highlight the take-home messages or lessons learned throughout the conducted work on acoustic event detection (AED) reaching out to open questions and future suggestions. For a smoother line of thought and because these points came out while working on one or the other problem, each of the following discussions will be organized per topic, *i.e.*, for note onset detection (NOD) and howling detection (HD) separately. Nevertheless, many of the remarks are valid for both and can be extended to other AED tasks, for instance the ones suffering from:

- limited dataset quality and availability,
- ambiguity or non uniqueness in target event definition,
- largely skewed classes distribution, *i.e.*, when the task is intrinsically having a large imbalance in the number of the event instances compared the the background.

9.1 Summary of contributions

After introducing the general of problem AED and the state-of-the-art solutions related to it, the presented work focused on two tasks in which the events are sharing a specific time-frequency pattern, namely NOD and HD. For each, the thesis proposed a novel detection feature, evaluation and training automatically annotated datasets and several suggestions for a more fair evaluation.

For the NOD task, we described a new way together with a MATLAB tool – Mix-Notes – for generating automatically annotated datasets, by first annotating the onsets for the notes when isolated then generating the mix with the corresponding annotation. Mix-Notes is equipped with various parameters to control different options of the generated excerpt. The resulting annotations are less subjective compared to the manually annotated datasets as they are human and note-context independent. We proposed two non-data-driven NOD features detecting onsets by measuring the signal's spectral sparsity, namely normalised identification of note onset based on spectral sparsity (NINOS^2) and identifying note onsets based on spectral sparsity (INOS^2). The latter is cheaper from a computational complexity point of view. Both features were preceded with a pre-processing step retaining only the low-energy frequencies. Both contributions, the feature(s) with the pre-processing resulted in a better performance for the challenging strings instrument having mainly soft – pitched non-percussive – onsets. Moreover, we proposed an evaluation metric, detections standard deviation, to assess how consistent is a feature in detecting onsets relative to each other. A study about a crucial and mostly missing evaluation parameter, the overall shift in annotations between different datasets, showed up to 50% increase in the reported F_1-score (F_1) which would drastically change the ranking of the compared algorithms. Finally, we suggested three augmentation strategies with semi-synthetic dataset for training the state-of-the-art data-driven NOD model which resuted in a higher and more stable performance for the piano use-case.

Inspired from the spectral sparsity features developed for NOD, we proposed NINOS^2-Transposed (NINOS^2-T) for the HD task. The novel feature is applied directly on the full spectrum, considering all frequency-time bins as howling candidates. This overcomes the limitations of the candidate-selection method used in earlier research and opens the door for complete howling detection. NINOS^2-T reported best average and worst-case performance, found the best compromise between precision and recall, and is less sensitive to threshold variance. This prominent news for the NINOS^2-T feature was for all the use-cases: full and early evaluation and for both music and speech examples. This novel feature is less prone to false positives (FPs) by depending on sparsity and not magnitude. This gives it the power to differentiate howling realizations from voiced speech and tonal music components. Moreover, it increases NINOS^2-T chances for an early howling detection or even when ringing is present. NINOS^2-T is intrinsically an online howling frequency detector. This is because the howling detection function (HDF) calculated using NINOS^2-T feature, is normalized in the range $[0, 1]$, hence the detection threshold choice is independent from the signal power. All of the other features - except for Interframe Peak Magnitude Persistence (IPMP) - needs a normalization step, *i.e.*, the range of the signal magnitude should be estimated and used to normalize

the HDF before applying the detection threshold. In the presented work this was done on the first 2 seconds of the snippet but in real applications, the normalization should be repeated more frequently which adds a certain delay in the detection procedure. For evaluating the performance of the proposed feature, we generated a large diversified dataset in which howling is simulated with a closed-loop feedback system. On one hand, this simulated-howling dataset is more realistic than the available synthetic datasets using sinusoidal superposition. On the other hand, it is less human dependent than manually annotated datasets as the only human contribution was in the post-processing pruning used to filter out few problematic excerpts (having multiple howling frequencies or no howling at all). For the evaluation, we used the well-known precision-recall (PR) curves together with its area under the PR curve (PR-AUC) instead of the receiver operating characteristic (ROC) and the area under the ROC curve (ROC-AUC). We showed experimentally how the latter were less adequate for the problem in hands due to the intrinsic high imbalance of the howling datasets, i.e., there are much more non-howling events (time-frequency bins) than the howling ones.

AED is a vast and rapidly growing field aiming to detect every possible event that is accompanied with sound emission. It suffices to look at the different detection challenges suggested in the yearly workshop on Detection and Classification of Acoustic Scenes and Events (DCASE) and the respective number of events to be detected. The tasks suggests detecting sounds in domestic environments (shouting, collapsing, falling down stairs, conversation, walking and screaming) [21, 22], anomalous machine sounds, urban sounds (cars parking, bikes, pedestrians) [19] and bird calls/songs (stress, pleasure, emergency). Only in DCASE, the event classes amounted for more than 700 [31] and this is only a very small portion of what can be detected and what could be beneficial to detect. Different AED tasks can benefit from the techniques suggested in this thesis. As mentioned in Section 2.4, the AED scheme – previously tailored to the NOD task – is applicable to many sound event detection tasks even for the state-of-the-art end-to-end models which can make use of it in their design and analysis. The idea behind the spectral sparsity measures (INOS^2, NINOS^2 and NINOS^2-T) or one of the measures themselves can be used to detect other types of acoustic events as transient property of event's starting point is quite common. For instance, the recent work suggesting the use of a related measure – spectral entropy – to detect the amount of birds vocalization [117]. Moreover, the overall shift in annotation is a crucial parameter for any music information retrieval (MIR) task that is time dependent and requiring higher detection resolution [118]. The manual annotation and its related shift in annotation problem doesn't concern MIR only but can be extended to any acoustic event.

9.2 Lessons learned

9.2.1 Note Onset Detection

- **Do not under-estimate the low-energy frequency components:** many of the non-data-driven features developed for NOD expected the information about onsets - or transients in practice - to be held in the musical notes' fundamental frequencies and harmonics. This was done explicitly by filtering them out as a pre-processing step, using for instance a western music scale filter, followed by feature calculation. Alternatively, this is done implicitly like with the Logarithmic Spectral Flux (LSF) in which the low-energy frequency components are masked by differences involving the high-energy ones, *i.e.*, fundamentals and harmonics. In our work, low-energy frequency components proved to hold enough knowledge about onsets, helping to reach a superior performance for the challenging sustained-strings instruments with soft onsets. The results were obtained by calculating the NINOS^2 feature exclusively on the low-energy bands and excluding the high-energy ones.

- **Many notes can hide in a music excerpt by not showing an energy increase, but almost all cannot hide that they needed some time to reach stability:** this is the main idea behind the development of NINOS^2 which expects each note as a baby starting from not knowing its personality, slowly discovering it and accentuating some of its marks. More precisely, the transient part of a note have its energy spread among the different frequencies, then the note reaches its steady-state where the energy is concentrated in its fundamental frequency and harmonics. This translates to a spectrally less sparse transient compared to a steady-state phase and hence is captured by NINOS^2 even for the challenging soft onsets.

- **It is hard to play hide-and-seek without knowing *Who* are we looking for. Before detecting onsets, we first need a clear and precise definition:** unfortunately, this kind of definition is not always possible. An onset is not a physical phenomenon but simply a time marker for transients. It could be the point in time when the player hits a piano key or when the piano note hits our ears. In earlier NOD solutions, onsets definition depended on available dataset annotation method which were mostly manual labeling. Based on an onset definition when notes are isolated and to reduce ambiguity, in this thesis we developed a tool for generating datasets having automatically annotated onsets. This results in

human and note-context independent annotation, hence no dependencies between training and testing examples, and with quasi-infinite availability.

- **The simple semi-synthetic augmentation is quite prominent for training deep data-driven models:** on one hand, this comes as a result of the semi-synthetic dataset properties: easy to generate, theoretically infinite and automatically annotated removing any human or context dependencies. On the other hand, the preliminary results for piano onset detection showed better and more stable performance when applying the suggested three augmentation strategies, *i.e.*, simultaneous and – partially & fully – freezing the feature-learning layers.

- **Training/Testing examples separation can be at different levels:** a straight forward and less strict strategy suggests dividing all of the generated semi-synthetic excerpts into two groups. One is used for training while the other for testing. A more strict strategy for challenging our learning algorithm would be to divide the isolated notes into two pools from the beginning. Here, one pool is used for generating the training while the other for the testing examples. In the presented work, we followed the latter strategy.

- **Datasets from different sources and labeling methods should be handled with care:** this is mainly because those datasets will implicitly assume different onsets definitions. This became obvious when comparing the results from three algorithms - NINOS^2 [33], LSF [37] and the Convolutional Neural Network (CNN) by [34] - on 4 different datasets: a semi-synthetic automatically annotated dataset , a real manually annotated dataset and two midi annotated disklavier datasets where one used ambient and the other used close recording setup. When studying the effect of the *overall shift in annotations* parameter introduced in Chapter 6, we see how the labels misalignment between the different datasets can result in a misleading evaluation - in some cases , reporting half of the real F_1. In NOD literature, this annotation mismatch was usually handled by using an evaluation window of 50ms within which a detection is considered a true positive (TP). The work presented in Chapter 6 showed that an evaluation window of at least 110ms is needed which would result in a quite bad detection resolution. This problem is magnified when dealing with Deep Neural Networks (DNNs), as the model generates very tight and sharp peaks around the detections. Hence, even a small shift is drastically affecting the results.

- **Data-driven NOD models are generally considered as the state-of-the-art solution, but resulted in worse performance when trained and tested on different datasets:** Added to the Music

Information Retrieval Evaluation eXchange (MIREX) challenge yearly results, the experiments using the CNN by [34] in Chapters 5 and 6 resulted mostly in the best performance. This is due to the higher complexity of a CNN model and its ability to model not only the onsets of individual notes but also their context. To achieve this kind of performance, the data-driven models need a large number of learnable parameters, large amount of data with high variability. This generally comes at the expense of using a considerable amount of training time and quiet higher memory and computaional complexity when compared to non-data-driven features. Moreover, as they are capable to learn much more from data than what they are expected to, techniques to avoid overfitting and adapttion to new datasets are unavoidable to maintain their ranking in the competition. This is supported experimentally in Chapter 6 where the CNN resulted in a lower performance compared to its non-data-driven rivals when applied on the string instruments excerpts from the semi-synthetic *PR8* dataset used in Chapter 4 while being trained on a manually annotated dataset – MDS_train – that contained more musically complex mixes. Finally, non-data-driven features are sometimes preferred to data-driven for some applications having restrictions on memory and computation resources and/or processing time.

9.2.2 Howling Detection

- **Howling and onsets detection are spectrally connected:** it could be luck, but this time it sufficed to rotate the spectrogram by 90° to notice the resemblance between onsets and howling at a certain frequency. Onsets are usually similar to vertical lines in a spectrogram while howling is a horizontal one in the neighboring of the resonant frequency. Onsets are spectrally non sparse in the frequency direction and howling is non sparse in the time direction.

- **Considering all frequencies as howling candidates takes the HD performance to another level:** we showed how earlier HD evaluation method assessed how well an algorithm do in differentiating the howling from the non-howling candidates. We claim this is missing a part of the HD story as it is limited to the candidates choice, *i.e.*, to the set of magnitude peaks chosen as candidates. Practically howling with lower energy than the signal's original components, will be excluded from the evaluation. For that reason, considering all frequency-time bins as candidates extends the performance comparison to early howling and ringing detection. Adopting this novel approach reports more realistic and fair comparison.

- **Counting TPs is tailored to the task in hand:** when tackling the NOD task, the features' performance was compared on F_1 basis applying the guidelines implied by the MIREX challenge. Practically, two detections within the ground-truth window, namely double detection, are considered as one TP and one FP. This relates to fact that NOD task uses larger evaluation windows (50ms) to handle lower precision in onsets definition and labeling and applies post-processing peak picking (PP) that combines nearby detections. In contrast, HD uses coarser evaluation windows (at most 2 neighboring frequency bins) to be in harmony with the notch-filter-based howling suppression (NHS) solution which aims for narrow-band filtering. For that reason, double detection are considered as a single TP.

- **Averaging F_1, ROC and PR could be tricky:** being the mostly used metrics and visualizations when assessing event detection algorithms, it should be clear what kind of average are we talking about. This issue relates to the micro and macro metrics averages defined for multi-classes problems. While macro calculates the metric independently for each of the classes and then take their average, micro will first aggregate the contributions of the different classes (TPs, FPs, true negatives (TNs), false negatives (FNs)) then compute the average metric. In the presented work, the same applies but on the excerpts and not the classes level, *i.e.*, the question is the reported metric a micro or macro average of the metric over the test examples. While the macro-F_1 treats all test examples equally, the micro-F_1 would be more exact by counting for the difference in examples' length. The same applies for the visual ROC and PR curves and their related area-under-the-curve and specially when curves interpolation is needed.

- **PR and PR-AUC are better performance evaluation tools when dealing with highly imbalanced classification problems and where TNs do not add value:** without zooming-in, ROC were useless to compare the performance of the different HD features and ROC-AUC values were quite misleading due to the large imbalance in the howling datasets. In contrast, by analyzing PR curves in Chapter 8, it was much easier to compare the features and study their sensitivity to threshold variance. The single-valued metric F_1 reports the performance for a specific threshold choice, differently from PR-AUC which is also a single-value metric but this time, assesses the features' performance independently from the threshold choice. One last thing to point out is the difference in the scale between ROC-AUC and PR-AUC as the baseline - for a randomly guessing algorithm (RGA) - for the first is constant and

equal to 0.5 while depending on the class skew and approaching 0 for highly skewed problems – 1:10 or less.

9.3 Future work

9.3.1 Note Onset Detection

- **Exploit the high-energy frequencies by fusing NINOS^2 with other non-data-driven features:** We find it promising to modify NINOS^2 by exploiting the information from the high-energy frequency components in addition to the low-energy ones. For this, sparsity-dependent features may be limited and fusion policies with features like LSF should be investigated. This is expected to lead not only to a performance improvement but possibly to generalize to a larger range of instruments. On the other hand, techniques are needed to make the dependency on low-energy components robust to noise when dealing with signals having low signal-to-noise ratio (SNR).

- **Towards a custom low-energy frequency coefficients percentage (γ) for a better performance per instrument, group of instruments and polyphony level:** As detailed in Section 4.4.1, the γ parameter defining the amount of low-energy frequency bins to be kept for the processing is set to 95.5%, a value found to maximise the F1-score over the tuning dataset (comprising excerpts from all the different instruments and playing styles). As a following step, some heuristics about the richness of harmonics would be valuable for producing customized onset detectors per instrument and per polyphony level.

- **Try multiple processing window sizes to account for different attack duration per instrument:** As NINOS^2 satisfies the *cloning* property of sparsity measures, different processing window sizes may be investigated to accommodate the different transient duration. This latter can be too short for percussive instruments (drums, piano, guitar) to very long for pitched non-percussive instruments (violin cello, accordion).

- **Further development for the Mix-Notes tool:** it can be focusing on intelligent music mixing by applying some music theory rules and enhancing the notes fade-out. Moreover, more challenging excerpts can be generated by mixing different instruments together and having excerpts densely populated with onsets for the instruments which are played with faster beats per minute (bpm).

- **Check the effect of semi-synthetic dataset proportion when augmentation is used with data-driven models:** As a follow-up of the work presented in Chapter 5, It would be interesting to assess the CNN's performance versus the proportion of semi-synthetic dataset size to real manually annotated one – in terms of training points. This should be done while considering the effect of model complexity, i.e the number of trainable parameters.

- **Interesting possibilities out of semi-synthetic augmentation for data-driven models:** The work presented in Chapter 5 can be extended to different instruments and more challenging mixes (excerpts with more onsets, adding noise and reverberation and mixing instruments together). But we are wondering if it is always beneficial to keep increasing the dataset size. In any case, this should be accompanied by model's capacity increase, for instance, increasing the number of nodes and layers for DNN. Note that increasing the dataset availability can be either by generating more excerpts, but also by recording and using more isolated notes which would ensure more data variability.

- **Overall shift in annotations can be used to align datasets used for models' training:** for more reliable and transparent evaluation, in Chapter 6, we suggested to report the F_1 as function of both thresholds and shifts. This issue is not only crucial for evaluation but also for training. The overall shift in annotation can be used to fix the mismatch between two or more datasets before using them for training. As mentioned previously, the idea of this shift parameter can be extended to any MIR time-dependent task.

- **The use of cross-dataset validation for a better generalization:** One could think of different levels of achievable generalization for the results depending on how datasets are involved in the validation step,

  - Single Dataset: the lowest level of generalisation where the evaluation is ran on a single dataset and the train/test examples are drawn from this dataset. The generalisation correlates only to the samples' variability in the used dataset.
  - Different Datasets: if similar results and conclusions are drawn for the different datasets separately, i.e. the train and test examples are drawn from the same dataset. On top of the per-dataset variability, the generalisation level correlates to the variability level (instruments, played songs/mixtures, notes, styles, recording, etc.) between the used datasets.

  - Cross-datasets: the highest and strongest generalisation could be achieved by training on one dataset while testing on another generating some sort of cross-dataset validation results.

In Chapter 5, we claim achieving the middle level in which two piano datasets were used but they differ in played songs and the recording setup (ambient and close). The cross-dataset validation should be adopted in future to strengthen the generalization [119].

9.3.2 Howling Detection

- **Explore fusion between NINOS^2-T and other features to increase robustness to FPs:** Despite the performance increase, there is still a lot to do and specially for the challenging music case. As a next step, fusion between different HD features can be tried out to increase their robustness.
- **Correct NINOS^2-T predictions using time series modeling:** One possibility is to use a Hidden-Markov model (HMM) or a Recurrent Neural Network (RNN) to model the howling evolution to correct the discontinuous NINOS^2-T detections to further differentiate howling from tonal music and voiced speech components.
- **Towards a better evaluation for early howling detection:** In the work presented in Chapter 8, an assessment of how early the howling frequency is detected is done by restricting the performance evaluation to the first few seconds of howling. A better alternative would be and assessment of the performance variability as a function of the time lag.
- **Class imbalance needs to be addressed in the training step and not only at the evaluation:** In this work we focused on the evaluation of imbalanced-classes problems. On the other hand, it is quite challenging to address the imbalance in learning and specially for deep learners. Several techniques are suggested in literature like down-sampling, up-sampling and weighting the cost function, with no proofs of convergence. This makes of it an attractive open research question.
- **PR analysis can be problematic depending on skew variability between datasets:** While being widely used for evaluating different classification tasks including MIR and more general AED due to its higher sensitivity to class imbalance, more recent research [57, 120] point out several limitations when compared the ROC analysis. In [120], the authors introduced the problem of unachievable region in PR curves and how this

can affect the visual assessment and the PR-AUC metric, and suggested a normalized version of the PR-AUC to count for the unachievable region. Similarly in [57], the authors described different appealing properties of the ROC analysis that are missing in the PR alternative while not neglecting the primacy of PR for highly skewed problems where the TNs do not add value. One of the main problems of PR analysis – like in [120] – is that PR-AUC is defined by an arithmetic mean which may reduce the correlation between its value and the harmonic mean F_1. This problem is more pronounced when comparing the performance over datasets with different skew ratio (training and testing datasets or several datasets or in a cross-validation setup) and especially when the skew is lower than the ones seen for NOD and HD. Nevertheless, for future HD and NOD evaluations, we should consider the different solutions suggested for the described problem. This can be the normalized PR-AUC [120], the precision-recall-gain (PRG) curve or the F-measure curves [121].

Bibliography

[1] S. K. Mitra, *Digital Signal Processing*, 4th ed. New York, NY: Wcb/McGraw-Hill, 2010.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, Dec. 2013.

[3] B. Gold, N. Morgan, and D. Ellis, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, Nov. 2011.

[4] H. M. Bolaños, *Introduction to Audio Analysis. A MATLAB Approach*. Academic Press, 2014.

[5] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, ser. Signals and Communication Technology. London: Springer International Publishing, 2015.

[6] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 2, pp. 356–370, Feb. 2012.

[7] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer International Publishing, Jul. 2015.

[8] P. Brossier, J. P. Bello, and M. D. Plumbley, "Real-time temporal segmentation of note objects in music signals," in *Proc. Int. Computer Music Conf.(ICMC '04)*, University of Miami, Coral Gables, Miami, USA, Aug. 2004.

[9] M. Freitas Caetano, J. J. Burred, and X. Rodet, "Automatic Segmentation Of The Temporal Evolution Of Isolated Acoustic Musical Instrument Sounds Using Spectro-Temporal Cues," in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx '10)*, Austria, Sep. 2010, pp. 11–21.

[10] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Brill Publishers, 2012.

[11] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A Tutorial on Onset Detection in Music Signals," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1035–1047, Sep. 2005.

[12] J. J. Valero-Mas, E. Benetos, and J. M. Iñesta, "Assessing the Relevance of Onset Information for Note Tracking in Piano Music Transcription," in *Proc. 2017 AES Int. Conf. on Semantic Audio*. Audio Engineering Society, Jun. 2017.

[13] C. Duxbury, M. Davies, and M. Sandler, "Separation Of Transient Information In Musical Audio Using Multiresolution Analysis Techniques," *Proc. COST G-6 Conf. Digital Audio Effects (DAFx '01)*, p. 4, 2001.

[14] T. van Waterschoot and M. Moonen, "Fifty Years of Acoustic Feedback Control: State of the Art and Future Challenges," *Proc. IEEE*, vol. 99, no. 2, pp. 288–327, Feb. 2011.

[15] ——, "Comparative Evaluation of Howling Detection Criteria in Notch-Filter-Based Howling Suppression," *J. Audio Eng. Soc.*, vol. 58, no. 11, p. 19, 2010.

[16] P. Leveau and L. Daudet, "Methodology and Tools for the evaluation of automatic onset detection algorithms in music," in *Proc. 5th Int. Symp. on Music Information Retrieval (ISMIR '04)*, 2004, pp. 72–75.

[17] T. Virtanen, M. D. Plumbley, and D. Ellis, "Introduction to Sound Scene and Event Analysis," in *Computational Analysis of Sound Scenes and Events*, T. Virtanen, M. D. Plumbley, and D. Ellis, Eds. Cham: Springer International Publishing, 2018, pp. 3–12.

[18] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, I. McLoughlin, and A. Mertins, "What makes audio event detection harder than classification?" in *Proc. 25th European Signal Process. Conf. (EUSIPCO '17)*, Aug. 2017, pp. 2739–2743.

[19] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in *Proc. 22nd ACM Int. Conf. on Multimedia*, ser. MM '14. New York, NY, USA: ACM Press, 2014, pp. 1041–1044.

[20] L. Lu and A. Hanjalic, "Text-Like Segmentation of General Audio for Content-Based Retrieval," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 658–669, Jun. 2009.

[21] Ya-Ti Peng, Ching-Yung Lin, Ming-Ting Sun, and Kun-Cheng Tsai, "Healthcare audio event classification using Hidden Markov Models and Hierarchical Hidden Markov Models," in *Proc. 2009 IEEE Int. Conf. on Multimedia and Expo (ICME '09)*, Jun. 2009, pp. 1218–1221.

[22] M. Nishida, N. Kitaoka, and K. Takeda, "Development and preliminary analysis of sensor signal database of continuous daily living activity over the long term," in *Signal and Information Process. Association Annual Summit and Conference (APSIPA)*, Asia-Pacific, Dec. 2014, pp. 1–6.

[23] C. S. Ooi, K. P. Seng, L.-M. Ang, and L. W. Chew, "A new approach of audio emotion recognition," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5858–5869, Oct. 2014.

[24] A. Temko, R. Malkin, C. Zieger, D. Macho, and C. Nadeu, "Acoustic event detection and classification in smart-room environments: Evaluation of CHIL project systems," *IV Jornadas en Tecnologia del Habla*, p. 6, 2006.

[25] R. Casati and A. Varzi, "Events," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., 2010.

[26] Xiaodan Zhuang, Xi Zhou, T. S. Huang, and M. Hasegawa-Johnson, "Feature analysis and selection for acoustic event detection," in *Proc. 2008 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '08)*, Mar. 2008, pp. 17–20.

[27] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, "HMM-Based Acoustic Event Detection with AdaBoost Feature Selection," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science, R. Stiefelhagen, R. Bowers, and J. Fiscus, Eds. Springer Berlin Heidelberg, 2008, pp. 345–353.

[28] Z. Huang, Y.-C. Cheng, K. Li, V. Hautamaki, and C.-H. Lee, "A Blind Segmentation Approach to Acoustic Event Detection Based on I-Vector," in *Proc. Interspeech '13*, 2013, p. 5.

[29] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Proc. 2011 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '11)*, Oct. 2011, pp. 69–72.

[30] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. V. hamme, "An exemplar-based NMF approach to audio event detection," in *Proc. 2013 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '13)*, Oct. 2013, pp. 1–4.

[31] X. Xia, R. Togneri, F. Sohel, Y. Zhao, and D. Huang, "A Survey: Neural Network-Based Deep Learning for Acoustic Event Detection," *Circuits Syst Signal Process*, vol. 38, no. 8, pp. 3433–3453, Mar. 2019.

[32] S. Böck and G. Widmer, "Maximum filter vibrato suppression for onset detection," in *Proc. 16th Int. Conf. Digital Audio Effects (DAFx '13)*, 2013.

[33] M. Mounir, P. Karsmakers, and T. van Waterschoot, "Guitar note onset detection based on a spectral sparsity measure," in *Proc. 24th European Signal Process. Conf. (EUSIPCO '16)*, Aug. 2016, pp. 978–982.

[34] J. Schluter and S. Bock, "Improved musical onset detection with Convolutional Neural Networks," in *Proc. 2014 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '14)*, May 2014, pp. 6979–6983.

[35] C. Duxbury, M. Sandler, and M. Davies, "A Hybrid Approach to Musical Note Onset Detection," *Proc. 5th Int. Conf. Digital Audio Effects (DAFx '02)*, p. 6, 2002.

[36] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Amer.*, vol. 103, no. 1, pp. 588–601, Jan. 1998.

[37] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proc. 1999 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '99)*, vol. 6, Mar. 1999, pp. 3089–3092 vol.6.

[38] S. A. Khoubrouy, I. M. S. Panahi, and J. H. L. Hansen, "Howling Detection in Hearing Aids Based on Generalized Teager-Kaiser Operator," *IEEE Trans. Audio Speech Lang. Process.*, pp. 1–1, 2015.

[39] X. Serra and J. Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[40] S. A. Khoubrouy and I. Panahi, "A method of howling detection in presence of speech signal," *Signal Processing*, vol. 119, pp. 153–161, Feb. 2016.

[41] S. Dixon, "Onset detection revisited," in *Proc. 9th Int. Conf. Digital Audio Effects (DAFx '06)*, 2006, pp. 133–137.

[42] A. Schloss, "On the automatic transcription of percussive music," Ph.D. dissertation, PhD thesis, Department of Speech and Hearing, Stanford University, 1985.

[43] C. Duxbury, M. Davies, and M. B. Sandler, "Improved Time-Scaling of Musical Audio Using Phase Locking at Transients," in *Preprints AES 112th Convention*. Audio Engineering Society, Apr. 2002.

[44] P. Masri, "Computer modelling of sound for transformation and synthesis of musical signals," Ph.D. dissertation, University of Bristol, 1996.

[45] S. Böck, F. Krebs, and M. Schedl, "Evaluating the Online Capabilities of Onset Detection Methods." in *Proc. 13th Int. Symp. on Music Information Retrieval (ISMIR '12)*, 2012, pp. 49–54.

[46] J. P. Bello and M. Sandler, "Phase-based note onset detection for music signals," in *Proc. 2003 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '03)*, vol. 5, Apr. 2003, pp. V–441.

[47] C. Duxbury, J. P. Bello, M. Davies, and M. Sandler, "A combined phase and amplitude based approach to onset detection for audio segmentation," in *Digital Media Processing for Multimedia Interactive Services*. World Scientific, Mar. 2003, pp. 275–280.

[48] J. Flocon-Cholet, J. Faure, A. Guerin, and P. Scalart, "A robust howling detection algorithm based on a statistical approach," in *Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC '14)*, Juan-les-Pins, France, Sep. 2014, pp. 65–69.

[49] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proc. 2016 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '16)*, Mar. 2016, pp. 6440–6444.

[50] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE Trans. Audio Speech Lang. Process.*, vol. 25, no. 6, pp. 1291–1303, Jun. 2017.

[51] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks." in *Proc. 11th Int. Symp. on Music Information Retrieval (ISMIR '10)*, 2010, pp. 589–594.

[52] E. Çakır and T. Virtanen, "End-to-End Polyphonic Sound Event Detection Using Convolutional Recurrent Neural Networks with Learned Time-Frequency

Representation Input," in *Proc. 2018 Int. Joint Conf. Neural Networks (IJCNN '18)*, May 2018.

[53] S. Sigtia, E. Benetos, and S. Dixon, "An End-to-End Neural Network for Polyphonic Piano Music Transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, May 2016.

[54] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1994.

[55] M. Tian, G. Fazekas, D. A. A. Black, and M. Sandler, "Design and Evaluation of Onset Detectors Using Different Fusion Policies," *Proc. 15th Int. Symp. on Music Information Retrieval (ISMIR '14)*, p. 6, 2014.

[56] S. Böck and G. Widmer, "Local Group Delay Based Vibrato and Tremolo Suppression for Onset Detection." in *Proc. 14th Int. Symp. on Music Information Retrieval (ISMIR '13)*, 2013, pp. 361–366.

[57] P. Flach and M. Kull, "Precision-Recall-Gain Curves: PR Analysis Done Right," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 838–846.

[58] T. Fawcett, "An introduction to ROC analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[59] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. 23rd Int. Conf. Machine Learning (ICML '06)*. Pittsburgh, Pennsylvania: ACM Press, 2006, pp. 233–240.

[60] D. Stowell and M. Plumbley, "Adaptive whitening for improved real-time audio onset detection," in *Proc. Int. Computer Music Conf.(ICMC '07)*, Aug. 2007, pp. 312–319.

[61] MIREX, "Onset detection results 2018," https://nema.lis.illinois.edu/nema_out/mirex2018/results/aod/summary.html, 2018.

[62] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. 2017 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '17)*, Oct. 2017, pp. 344–348.

[63] M. Rossignol, G. Lafay, M. Lagrange, and N. Misdariis, "SimScene : A web-based acoustic scenes simulator," in *1st Web Audio Conference (WAC)*. Paris, France: IRCAM & Mozilla, Jan. 2015.

[64] F. Opolko and J. Wapnick, *McGill University Master Samples*. Montreal, QC, Canada: McGill University, Oct 2006, DVD edition.

[65] M. Mounir, "Note onset detection using sparse over-complete representation of musical signals," Master's thesis, Advanced Learning and Research Institute - USI, 2013. [Online]. Available: ftp://ftp.esat.kuleuven.be/stadius/mshehata/mscthesis/mshehatamsc.pdf

[66] E. Benetos and S. Dixon, "Polyphonic music transcription using note onset and offset detection," in *Proc. 2011 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '11)*, May 2011, pp. 37–40.

[67] T. Cheng, M. Mauch, E. Benetos, and S. Dixon, "An attack/decay model for piano transcription," *Proc. 17th Int. Symp. on Music Information Retrieval (ISMIR '16)*, Aug. 2016.

[68] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri, "Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms," *Journal of New Music Research*, vol. 36, no. 1, pp. 1–16, Mar. 2007.

[69] B. D. Giorgi, M. Zanoni, S. Böck, and A. Sarti, "Multipath Beat Tracking," *J. Audio Eng. Soc.*, vol. 64, no. 7/8, pp. 493–502, Aug. 2016.

[70] P. Masri and A. Bateman, "Improved Modelling of Attack Transients in Music Analysis-Resynthesis," in *Proc. Int. Computer Music Conf.(ICMC '96)*, 1996, pp. 100–103.

[71] V. Verfaille, U. Zölzer, and D. Arfib, "Adaptive digital audio effects (a-DAFx): a new class of sound transformations," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 5, pp. 1817–1831, Aug. 2006.

[72] O. Celma, *Music Recommendation and Discovery.* Springer, 2010.

[73] N. Degara, M. E. P. Davies, A. Pena, and M. D. Plumbley, "Onset Event Decoding Exploiting the Rhythmic Structure of Polyphonic Music," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1228–1239, Oct. 2011.

[74] S. Böck, A. Arzt, and F. Krebs, "Online Real-time Onset Detection with Recurrent Neural Networks," *Proc. 15th Int. Conf. Digital Audio Effects (DAFx '12)*, p. 4, 2012.

[75] "Mirex 2015 onset detection results," http://nema.lis.illinois.edu/nema_out/mirex2015/results/aod/, 2015, accessed 2015-12-08.

[76] C. Liang, L. Su, and Y. Yang, "Musical Onset Detection Using Constrained Linear Reconstruction," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 2142–2146, Nov. 2015.

[77] N. Kroher and E. Gómez, "Automatic Transcription of Flamenco Singing From Polyphonic Music Recordings," *IEEE Trans. Audio Speech Lang. Process.*, vol. 24, no. 5, pp. 901–913, May 2016.

[78] N. Hurley and S. Rickard, "Comparing Measures of Sparsity," *IEICE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4723–4741, Oct. 2009.

[79] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A New Python Audio and Music Signal Processing Library," in *Proceedings of the 2016 ACM on Multimedia Conf.*, ser. MM '16. New York, NY, USA: ACM Press, 2016, pp. 1174–1178.

[80] A. Lerch and I. Klich, "On The Evaluation Of Automatic Onset Tracking Systems," zplane.development, Techreport, Jan. 2005.

[81] A. Roebel, C. Jacques, and A. Aknin, "MIREX 2018: Training CNN onset detectors with artificially augmented datasets," UMR STMS - IRCAM, CNRS, Paris Sorbonne University, Tech. Rep., 2018.

[82] Y. Mu, W. Ding, and D. Tao, "Local discriminative distance metrics ensemble learning," *Pattern Recognition*, vol. 46, no. 8, pp. 2337 – 2349, 2013.

[83] V. Emiya, "MAPS database: A piano database for multipitch estimation and automatic transcription of music," http://www.tsi.telecom-paristech.fr/aao/en/2010/07/08/maps-database-a-piano-database-for-multipitch-estimation-and-automatic-transcription-of-music/, 2008.

[84] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010.

[85] M. Mounir, P. Karsmakers, and T. van Waterschoot, "Annotations Time Shift: A Key Parameter in Evaluating Musical Note Onset Detection Algorithms," in *Proc. 2019 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '19)*, New Paltz, NY, USA, Oct. 2019, pp. 21–25.

[86] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. 3rd Int. Conf. on Learning Representations (ICLR '15)*, San Diego, CA, USA, May 2015.

[87] H. Schreiber and M. Müller, "Accelerating Index-Based Audio Identification," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1654–1664, Oct. 2014.

[88] A. L.-C. Wang, "An Industrial-Strength Audio Search Algorithm," in *Proc. 4th Int. Symp. on Music Information Retrieval (ISMIR '03)*, 2003, pp. 7–13.

[89] J. Haitsma and T. Kalker, "A Highly Robust Audio Fingerprinting System With an Efficient Search Strategy," *Journal of New Music Research*, vol. 32, no. 2, pp. 211–221, Jun. 2003.

[90] J. Salamon and J. Urbano, "Current Challenges in the Evaluation of Predominant Melody Extraction Algorithms." in *Proc. 13th Int. Symp. on Music Information Retrieval (ISMIR '12)*, vol. 12, 2012, pp. 289–294.

[91] D. Thomas and A. R. Jayan, "Automated suppression of howling noise using sinusoidal model based analysis/synthesis," in *Proc. 2014 IEEE Int. Adv. Computing Conference (IACC '14)*, Gurgaon, India, Feb. 2014, pp. 761–765.

[92] D. T. Murphy and S. Shelley, "Openair: An interactive auralization web resource and database," in *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.

[93] J. Kearns, "Librivox: Free public domain audiobooks," *Reference Reviews*, 2014.

[94] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A DATASET FOR MUSIC ANALYSIS," *Proc. 18th Int. Symp. on Music Information Retrieval (ISMIR '17)*, p. 8, 2017.

[95] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process. Mag.*, vol. 9, no. 1, pp. 14 –37, Jan. 1992.

[96] P. R. Williams, "Method and system for elimination of acoustic feedback," U.S. Patent Application 2010/0 046 768 A1, Feb., 2010.

[97] M. H. Er, T. H. Ooi, L. S. Li, and C. J. Liew, "A DSP-based acoustic feedback canceller for public address systems," in *Proc. Int. Conf. Signal Process. (ICSP '93)*, Beijing, China, Oct. 1993, pp. 1251–1254.

[98] ——, "A DSP-based acoustic feedback canceller for public address systems," *Microprocessors and Microsystems*, vol. 18, no. 1, pp. 39–47, Jan./Feb. 1994.

[99] P. R. Williams, "Method and system for elimination of acoustic feedback," WIPO Patent Application WO/2002/021 817, Mar., 2002.

[100] M. Hanajima, M. Yoneda, and T. Okuma, "Howling eliminator," WIPO Patent Application WO/1999/021 396, Apr., 1999.

[101] ——, "Howling eliminating apparatus," U.S. Patent 6,125,187, Sep., 2000.

[102] A. Kawamura, M. Matsumoto, M. Serikawa, and H. Numazu, "Sound amplifying apparatus with automatic howl-suppressing function," U.S. Patent 5,442,712, Aug., 1995.

[103] ——, "Sound amplifying apparatus with automatic howl-suppressing function," European Patent Application EP0 599 450 A2, Jun., 1994.

[104] A. F. Rocha and A. J. S. Ferreira, "An accurate method of detection and cancellation of multiple acoustic feedbacks," in *Preprints AES 118th Convention*, Barcelona, Spain, May 2005, AES Preprint 6335.

[105] M. Börsch, "Method for constraining electroacoustic feedback," European Patent Application EP1 684 543 A1, Jul., 2006.

[106] ——, "Method for suppressing electro-acoustic feedback," U.S. Patent Application 2006/0 159 282 A1, Jul., 2006.

[107] D. Somasundaram, "Feedback cancellation in a sound system," European Patent Application EP1 903 833 A1, Mar., 2008.

[108] ——, "Feedback cancellation in a sound system," U.S. Patent Application 2008/0 085 013 A1, Apr., 2008.

[109] S. Ando, "Howling detection and prevention circuit and a loudspeaker system employing the same," U.S. Patent 6,252,969, Jun., 2001.

[110] N. Osmanovic and V. Clarke, "Acoustic feedback cancellation system," U.S. Patent Application 2007/0 019 824 A1, Jan., 2007.

[111] ——, "Acoustic feedback cancellation system," WIPO Patent Application WO/2007/013 981, Feb., 2007.

[112] ——, "Acoustic feedback cancellation system," U.S. Patent 7,664,275, Feb., 2010.

[113] D. M. Oster, M. P. Lewis, and T. J. Tucker, "Method and apparatus for adaptive audio resonant frequency filtering," WIPO Patent Application WO/1991/020 134, Dec., 1991.

[114] M. P. Lewis, T. J. Tucker, and D. M. Oster, "Method and apparatus for adaptive audio resonant frequency filtering," U.S. Patent 5,245,665, Sep., 1993.

[115] E. T. Patronis, Jr., "Acoustic feedback detector and automatic gain control," U.S. Patent 4,079,199, Mar., 1978.

[116] ——, "Electronic detection of acoustic feedback and automatic sound system gain control," *J. Audio Eng. Soc.*, vol. 26, no. 5, pp. 323–326, May 1978.

[117] K. A. Herborn, A. G. McElligott, M. A. Mitchell, V. Sandilands, B. Bradshaw, and L. Asher, "Spectral entropy of early-life distress calls as an iceberg indicator of chicken welfare," *Journal of The Royal Society Interface*, vol. 17, no. 167, p. 20200086, Jun. 2020.

[118] A. Pankajakshan, H. Bear, and E. Benetos, "Onsets, Activity, and Events: A Multi-task Approach for Polyphonic Sound Event Modelling," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE '19)*, New York, NY, USA, Oct. 2019, pp. 174–178.

[119] A. Livshin and X. Rodex, "The Importance of Cross Database Evaluation in Sound Classification," in *Proc. 4th Int. Symp. on Music Information Retrieval (ISMIR '03)*, Baltimore, United States, Oct. 2003, pp. 1–1.

[120] K. Boyd, V. Santos Costa, J. Davis, and C. D. Page, "Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation," in *Proc. 29th Int. Conf. Machine Learning (ICML '12)*, vol. 2012, Dec. 2012, p. 349.

[121] R. Soleymani, E. Granger, and G. Fumera, "F-measure curves: A tool to visualize classifier performance under imbalance," *Pattern Recognition*, vol. 100, p. 107146, Apr. 2020.

Curriculum Vitae



Mina Mounir obtained his B.S. in computer engineering from Cairo University, Egypt, in 2008. Between 2009 and 2011, he gained some industry experience mainly in the Automotive and IT fields. In 2013, he received his Msc. degree in Embedded Systems Design from Università della Svizzera Italiana (USI), Switzerland. From 2014 to 2020, he has been a doctoral researcher at KU Leuven, Belgium. His research is focused on acoustic event detection. This includes the design of handcrafted and data-driven features/models, automatically-annotated dataset generation and more fair evaluation metrics. He served as a reviewer for the International Society of Music Information Retrieval (ISMIR).

Publication List

International Journal Papers

1. M. Mounir, P. Karsmakers, and T. van Waterschoot, "Musical note onset detection based on a spectral sparsity measure," submitted for publication.

2. M. Mounir, G. Bernardi, and T. van Waterschoot, "Robust and early howling detection based on a sparsity measure," submitted for publication.

International Conference Papers

1. M. Mounir, P. Karsmakers, and T. van Waterschoot, "CNN-based Note Onset Detection using Synthetic Data Augmentation," in Proc. 28th European Signal Process. Conf. (EUSIPCO '20), Amsterdam, Netherlands, Aug. 2020.

2. M. Mounir, P. Karsmakers, and T. van Waterschoot, "Annotations Time Shift: A Key Parameter in Evaluating Musical Note Onset Detection Algorithms," in *Proc. 2019 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '19)*, New Paltz, NY, USA, Oct. 2019, pp. 21–25.

3. M. Mounir, P. Karsmakers, and T. van Waterschoot, "Guitar note onset detection based on a spectral sparsity measure," in Proc. 24th European Signal Process. Conf. (EUSIPCO '16), Aug. 2016, pp. 978–982.



FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING
Stadius Center for Dynamical Systems,
Signal Processing and Data Analytics
Kasteelpark Arenberg 10, B-3001 Leuven
http://www.esat.kuleuven.be/stadius/