

A Machine Learning Approach to the Perception of Phrase Boundaries in Music

Senior Project Submitted to
The Division of Science, Math, and Computing
of Bard College

by
Evan Petratos

Annandale-on-Hudson, New York
December 2020

Abstract

Segmentation is a well-studied area of research for speech, but the segmentation of music has typically been treated as a separate domain, even though the same acoustic cues that constitute information in speech (e.g., intensity, timbre, and rhythm) are present in music. This study aims to sew the gap in research of speech and music segmentation. Musicians can discern where musical phrases are segmented. In this study, these boundaries are predicted using an algorithmic, machine learning approach to audio processing of acoustic features. The acoustic features of musical sounds have localized patterns within sections of the music that create aurally perceptible “events” that musicians identify as distinctive characteristics of a phrase. An experiment was conducted to gather data from musicians for the machine learning algorithm, and to set an upper bound on the performance of such an algorithm. The algorithm succeeded in detecting phrase boundaries, as determined by the participants, with accuracy scores of 0.91, 0.67, and 0.60 for the data from three participants, but there are still improvements to be made--specifically, the low specificity of the machine learner’s prediction is a challenge for a future endeavor.

Acknowledgements

I would like to thank my advisor, Sven Anderson, for his relentless support and enthusiasm for this research, for his guidance and perspective on the research process, and for instilling a love for academia that will last a lifetime.

I would also like to give a special thanks to:

- my family, my friends, and my girlfriend, Brianna, for their persistent encouragement throughout this project.
- my musical mentors, Derek Fenstermacher and Marcus Rojas, for their guidance and patience over the years.
- my conservatory advisor, Raman Ramakrishnan, for sharing his knowledge of time management.
- Zach, for being there every step of the way.

As well as:

- Matthew Deady for lending me direction and a valuable resource, Roederer's *The Physics and Psychophysics of Music*.
- Keith O'Hara, Thomas Hutcheon, and Erica Kiesewetter for reading this project.

Contents

1 Introduction.....	1
1.1 Literature Review.....	2
1.2 Link to Speech.....	7
1.3 Phrases.....	8
2 Methods.....	9
2.1 Experiment.....	10
2.1.1 Participants.....	10
2.1.2 Experimental Procedure.....	10
2.2 Analytical Methods.....	15
2.2.1 Praat.....	15
2.2.2 Librosa.....	16
2.2.3 Acoustic Features.....	17
2.2.4 Phrase Detection.....	22
2.3 Machine Learning.....	24
2.3.1 Logistic Regression.....	25
2.3.2 Feedforward Neural Network.....	27
3 Results.....	31
3.1 Participant Responses.....	32
3.1.1 Overview.....	32
3.1.2 Participant Phrase Markings.....	32
3.2 Participant Reliability.....	37
3.2.1 Precision/Recall Metrics and the F1 Measure.....	38
3.2.2 Phrase Lengths.....	41
3.3 Phrase Detection Algorithm.....	43
3.3.1 Acoustic Features as Predictors.....	43
3.3.2 Acoustic Features compared with Participants.....	45
3.4 Machine Learning Models.....	46
3.4.1 Logistic Regression Model.....	47
3.4.2 Feedforward Neural Network Model.....	52
4 Discussion.....	57
4.1 Evaluation.....	58
4.1.1 Experiment.....	58
4.1.2 Algorithmic Evaluation.....	60
4.1.3 Machine Learning Evaluation.....	61
4.2 Future Work.....	64
4.2.1 Improvements to the Machine Learning Models.....	65

4.2.2 Possible Application.....	67
4.3 Conclusion.....	67
Bibliography.....	69
Appendices.....	75
Appendix 1 Experiment.....	76
Appendix 1A IRB Approval Letter.....	76
Appendix 1B Verbal Instructions for Experiment.....	77
Appendix 1C Consent Form.....	78
Appendix 1D Participant Questionnaire.....	80
Appendix 1E Recruitment Email.....	82
Appendix 1F CITI Program Human Subject Research Training Certificate.....	83
Appendix 1G Debriefing Statement.....	84
Appendix 2 Plots of Acoustic Features.....	86
Appendix 2A Logarithm of Intensity Plots.....	86
Appendix 2B Acoustic Onsets Plots.....	87
Appendix 2C Spectrograms.....	89
Appendix 2D Marked Intensity Plots.....	89
Appendix 2E Marked Spectral Flatness Plots.....	91
Appendix 2F Marked Rhythmic Density Plots.....	92

1. Introduction

Phrasing in music, the way in which a musical passage is expressed to relay auditory information, is thought to be subjective (Olsen, 2016). The artistic decisions that drive a musician to create a phrase a certain way, with a certain length, is always at the discretion of that musician. The musical phrase is a means of expression, similar to language. Human language has a natural rise and fall in its contour of loudness, pitch, and rhythm, which conveys meaningful information (Olsen, 2016; Knösche, 2005; Roederer, 2008). Language is formed with words, clauses, sentences, etc., and can have inflections, pauses, articulation, and mood (Olsen, 2016). All of these attributes of language are used to enunciate and articulate the speaker's message. Many, or perhaps all, of these attributes are integral in the expression of a musical message (Roederer, 2008). Music and speech are related in this way (Knösche, 2005), and the previous endeavors on the analysis of human language should be considered in the analysis of musical expression. Phrase boundaries have proven to be discernible and identifiable in speech (Knösche, 2005; Glushko, 2016), but there is much less study, and published research concerning phrasing

in music. This project aims to define phrase boundaries in music--more specifically, phrase boundaries in the music of the Western classical tradition.

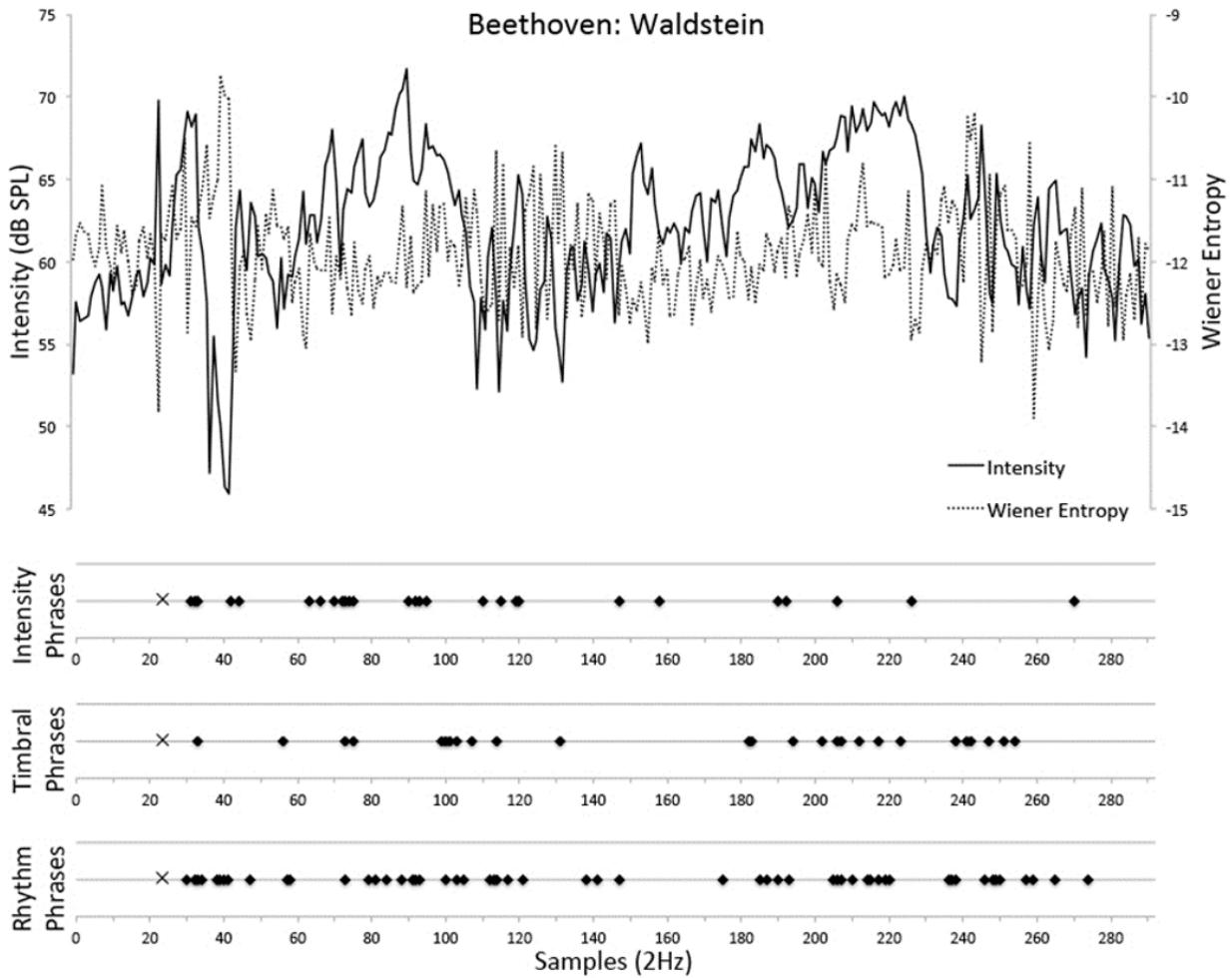
One goal of this project is to create a working, efficient algorithm that can detect phrase boundaries in music from a given audio file. The acoustic features involved in detecting phrases from a given audio file include (1) waveform analysis of the square of amplitude or loudness, called intensity, (2) spectral analysis of timbre using the tonality coefficient, spectral flatness, and (3) an analysis of onset detection, by measuring rhythmic density with half-second frames. In each case, the goal is to look for common patterns that occur between “events” during an audio file (Olsen, 2016). For data collection and correctness of the algorithm, segmentation of the sound into phrases is determined by human participants who intently listen to certain audio files and mark where they perceive the phrases end. In the analysis, the three aforementioned analysis techniques are considered and their accuracy in predicting event segmentation analyzed. These trends of the data from the participants are taken into account for the algorithm that dictates a prediction of the endings of such events.

Section 1.1: Literature Review

The largest portion of my advances in background for the project has been from Olsen’s paper, “What Constitutes a Phrase in Sound-Based Music?” (Olsen, 2016). The paper discusses how phrasing organizes auditory information in speech and music. For humans discerning boundaries in speech segments (particularly words, clauses, and sentences), there are well-studied acoustic cues that underlie the ability. These acoustic cues are realized as changes in intensity/amplitude, rhythmic pattern, and pitch contour. When we think about human speech,

there are natural inflections that occur to create distinct words and sentences. The same acoustic cues that determine these natural inflections in speech are found in sound-based music and note-based music (Olsen, 2016).

Olsen used the Beethoven Waldstein Piano Sonata as an example of note-based music, since the piece consists of many repeated, single-note motifs. Figure 1 shows the intensity, and spectral flatness (Wiener Entropy) accompanied with the participants' perceived phrase boundaries below the plot. It is inferred that the samples on the x-axis occur two times per second, since the excerpt they use is 146 seconds.



From Figure 1 in Olsen's paper. Time-stamped phrase responses and acoustic time-series data. The figure displays all time-stamped responses assigned to timbre, intensity, or rhythm categories across the entire time-course for Beethoven's Waldstein. Acoustic categories were based on thematic analyses of participants' qualitative descriptions of each perceived phrase. The diamonds mark where the subject perceived the end of a phrase. The 'x' symbol shows where the subject was instructed to begin marking where phrases end. Time-series data of acoustic intensity (solid line; dB SPL on left y-axis) and spectral flatness (dashed line; Wiener entropy on right y-axis) are also plotted.

Sound-based music, as described in Olsen's paper, is different from noted-based music where the basic unit from which the music is created is commonly non-instrumental, uninterrupted sounds, rather than instrumental, discrete notes. Olsen finds that phrases in

sounds-based music are more easily perceived by changes in intensity or timbre, whereas phrases in note-based music are better discerned with compressions and elongations in rhythmic density.

Other important predictors for note-based music are articulations, sustain, and decay of notes.

Another important source for this project is (Knösche, 2005). The paper is motivated by the notion that speech and music follow an uneven acoustic flow. The flow is partitioned into structures which can be interpreted as phrases. Phrase boundaries become a central component for the perception of these structures. Knösche's study used EEG and MEG to find correlations for the perception of phrase boundaries in music. They found that the timing and structure of the EEG and MEG data struck a resemblance to the phrase boundaries of prosodic speech from an earlier study. Such a discovery had a profound impact on this project, as a different angle emerged: the consideration of segmentation in human speech directly correlates to phrase boundaries in music.

In (Gingras, 2016) the idea was to analyze the relationships among predictability in musical structure (musical phrasing for the purposes of the present study), a performer's expressive timing (elongating or compressing tempi), and the listeners' perceived musical tension. In Gingras's study, melodic expectation was measured using a probabilistic model which was previously compared to that of human listeners. Gingras proposes that listeners' perceived musical tension is directly related to the performer's fluctuation in tempo. It is understood that musical tension could be predicted given a performer's expressive timing, thus strengthening the link between melodic expectation and tempo fluctuation in music.

Another important paper pertaining to this project has been (Glushko, 2016). Glushko studies a specific event-related potential component, the Closure Positive Shift (CPS), which is a

neural measure of phrase boundary perception. In previous studies, there has been separation between the CPS of language (language-CPS) and the CPS of music (music-CPS), which tells us where phrase boundaries are perceived for each medium. Prior studies assert that the music-CPS differ substantially from the language-CPS. However, contrary to these studies, Glushko finds a positive shift of musical phrase boundaries that strongly resembles the language-CPS. Another angle in this study, relating to the link between music and language, was to compare both the language-CPS and the music-CPS between musicians and non-musicians. The study found that the language-CPS in musicians was less pronounced than in non-musicians, suggesting more efficient processing in prosodic phrasing from higher musical expertise. The implications of the CPS in music and in language for discerning phrase boundaries are also discussed in (Silva 2014).

From reading multiple sources pertaining to phrasing in music, a few central themes emerged that became central to this project. First, it is natural for humans to perceive “events” in music or language (Olsen, 2016; Knösche, 2005; Glushko, 2016). In each study, the existence of boundaries in music and language was expected, and explored by different methods. In (Olsen, 2016), subjects listened to six different forms of musical stimuli and were asked to mark a boundary when the subject perceived an “event” had occurred. In (Knösche, 2005), subjects were monitored using EEG and MEG with the intention of finding a resemblance in the location of “events” in music and prosodic speech. Glushko’s study (Glushko, 2016) focused primarily on event boundaries in language and music, using EEG to locate such “events” and compared the location of where the boundary was perceived in language versus where the boundary was perceived in music. “Event” boundaries in each case translate to phrase boundaries; a point to be

made is that the boundaries of phrases were of utmost importance for the analysis of how humans interpret information in language and in music (Olsen, 2016; Roederer, 2008).

Section 1.2: Link to Speech

One critical observation among these studies is the link between speech prosody and music (Olsen, 2016; Knösche, 2005; Glushko, 2016; Roederer, 2008). Both speech and music carry acoustic information which can be interpreted by the human brain (Roderer, 2008). It is perhaps obvious that the sound produced from human speech carries information, since speaking is something humans develop as a means to relay information in messages and in conversation. But, music can be considered the co-product of the evolution of human language (Roederer, 2008). In the evolution of hominids, operations of sound processing, analysis, storage, and retrieval became necessary for the development of human speech. Such advancements progressed the reception of music and the perception of subjective sensations of timbre, consonance, tonal expression, sense of resolution, and the long-term structures of melodic lines. The perception of these sensations is linked to limbic rewards in the search for phonetic content of sound that can be identified as logical manifestations of acoustical signals (Roederer, 2008). The limbic system works in sort of a “binary” way; it dispenses either reward or punishment, which are emotional states of the brain. The motivation to listen to, analyze, and store musical sounds, even when there is no apparent circumstantial need, triggers a feeling of pleasure--this limbic reward. To facilitate information processing in speech and in music, the motivation emerged to understand acoustical signals and receive emotional feedback (Roederer, 2008). That is to say, we hear and interpret music in the same way that we hear and interpret speech.

Section 1.3: Phrases

Previously in this chapter, the term “events” was used to mean phrases. But since we are aiming to define events in music and in speech, the initial language we use to discuss these events should be more general (Olsen, 2016).

Through a vast collection of studies on speech and studies on speech segmentation, we know much about how events in speech are organized into information (Olsen, 2016; Knösche, 2005; Roederer, 2008). And, drawing upon learned knowledge from the previously aforementioned studies, we know speech and music both follow a certain, uneven acoustic flow, and this flow is partitioned into structures that we can identify as these events (Knösche, 2005). Since these events are well known in the study of speech segmentation, and both speech and music follow a similar pattern in their structure in organizing information, we can make the case that the events described above should be interpreted as phrases.

Phrase Endings

In this study, the focus will be primarily on the ends of phrases. The starting point of the next phrase, theoretically, will occur at the same moment that the previous phrase ends. This would not necessarily always be the case if the sound stops at the ending of the previous phrase--in this case, the starting point of the next phrase follows the silence. However, for most sound-base music, the ending of the previous phrase always marks the beginning of the next phrase. Since the pieces explored in this study are mostly continuous and uninterrupted by silence, the endings of phrases will be the objects to be discerned and studied.

2. Methods

The experiment described in this chapter was designed specifically to obtain phrase boundary markings from musicians. This experiment took inspiration from the experiment conducted in (Olsen, 2016), in which participants (mostly nonmusicians) listened to six stimuli, twice through, marking the endings of perceived “events,” and described the reasoning for their marking of each particular phrase ending. The aim of that experiment was to learn what the participants described as constituting a phrase in music. This experiment’s aim was to simply gather musicians’ phrase boundary markings--however the participants perceived a phrase boundary was at the discretion of the participants. The program that ran the experiment was written in Python using the Psychopy module [21]. The analysis of this data was conducted using a few Python modules: Librosa [14], SciPy [17], NumPy [18], and Matplotlib [19]. The machine learning models were implemented using Scikit-learn [15] and Tensorflow [16]. The methods and procedures for this experiment and for the analysis of the data are outlined below.

Section 2.1: Experiment

2.1.1 Participants

Sixteen undergraduate students studying music at Bard College's Conservatory of Music were recruited to participate in the experiment. Before starting the experiment, the participants were asked to complete a brief questionnaire about any hearing impairments they may have, their musical background, and their familiarity with concepts of physics and machine learning. The participants' instructions and questionnaire can be found in Appendix 1. All procedures involving human participants were conducted in accordance with the ethical standards of Bard College's Institutional Review Board (Case number: 2020OCT20-PET) and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. Each participant provided written, informed consent.

2.1.2 Experimental Procedure

Preliminary Procedures

Each participant was verbally briefed on how their data will be recorded without disclosing that their data will be used to train a computer program. They were then presented with a formal consent form, which also details COVID-19 safety procedures that were followed during the experiment. Upon agreeing to participate in the experiment, the participants were given a brief questionnaire, asking questions related to their objective hearing abilities, musical

background, personal tastes in music, and background in computation and acoustics. The results of the questionnaire are discussed in Section 3.1.1.

For the experiment, each participant was presented with ten excerpts, played twice, from the six stimuli (data from different parts of the excerpts is used as test data¹). The excerpts differ in style but are all of the Western Classical tradition. Some of the styles that were included are piano piece, large orchestra, chamber ensemble, and vocal music. The stimuli were presented in a public, open area (to allow for physical distancing) in the Reem and Kayden Center for Science and Computation via my personal laptop (HP Pavilion Notebook) and professional studio headphones (AKG K240 Studio over-ear headphones), which were sanitized between participants. The volume of the headphones was adjusted at the participants' discretion as to decrease any possible discomfort during the experiment.

Stimuli

Ten excerpts from six pieces of the Western Classical Tradition were used for the experiment.

1. Gregorian Chant, *Hymnus* [22] (the first 1'40"). This piece involves a choir singing in monophonic unison. The phrase structure is quite obvious because of the unison breaths between each phrase. This piece was used primarily as a base case for the identification of phrase boundaries.

¹ The test data comes from the same pieces, but at a different place, to check the accuracy of the algorithm. It is worth considering that this practice of using the same pieces as testing data could also lead to making inadequate predictions on other pieces, as the test data could be too similar to the training data and thus, the algorithm would not perform as well in the general case (i.e., the algorithm could yield poor predictions on other pieces).

2. Wolfgang Amadeus Mozart, *Symphony No. 40 in G minor, K. 550* [23] (the first 1'23" and another section 1'13"). This piece involves an orchestra with a strings section and several wind instruments. The phrase structure is not particularly obvious, so the perceived phrase boundaries differed in length from listener to listener. There are moments where certain sections of the orchestra play motifs at different points in the phrase. Both excerpts are from the first movement, *Molto Allegro*.
3. Ludwig van Beethoven, *Sonata No.21 in C Major, "Waldstein"* [24] (the first 2'26"). The piece involves solo piano. The piece has some minor tempo fluctuations, but is ultimately rhythmically driven. Some distinctions in phrasing were drawn when the rhythmic profile digressed to something different from what had previously occurred. The excerpt is from the first movement, *Allegro con brio*.
4. Anton Bruckner, *Symphony No.7 in E major, WAB 107* [25] (the first 2'16", another section 1'41", and another section 1'25"). This piece involves a large orchestra with a full strings section, winds, brass, and percussion. All three excerpts are from the first movement, *Allegro moderato*.
5. Frédéric Chopin, *Ballade no. 1 in G minor, op. 23* [26] (the first 1'57" and another section 1'15"). This piece involves solo piano. The piece has an obvious human component to the music, using much rubato and elongated motifs. Rather than relying on rhythm, as in the other piano piece on this list, the dynamics and tempo fluctuations generally indicated a phrase's direction and subsequent ending.
6. Sergei Rachmaninoff, *Vocalise Op. 34 No. 14* [27] (the first 1'29"). This piece involves cello and piano in a chamber setting. The piece was written in the 20th century and uses

less conventional harmonies than the other pieces explored here. The dynamic stays fairly uniform throughout the piece, so there are other harmonic aspects that can help determine the structure of the phrases.

The stimuli were always presented in the order indicated above, with all sections from the same piece playing before moving on to the next. It is worth noting that each stimulus section was presented twice to each participant, one after the other, to give the participants a chance to become more familiar with the style and perceived phrase lengths (i.e., the first Mozart section twice, then the second Mozart section twice, then the Beethoven excerpt twice, etc.). The participants tended to appreciate the second chance of getting to mark phrases on the second listen, so they could revisit the possible mistakes they felt that they made during the first listen.

The program that presented the stimuli displayed written instructions detailing how to use the program to mark the phrases. When the participants began, the instructions stayed on the screen, so there was no confusion throughout the experiment.

Experimental Procedure

Each participant intently listened to each section of the excerpts two times through, consecutively. The first listen permitted the participant to become familiar with the style and the perceived phrase lengths (this was fully disclosed to the participants before beginning the experiment). Before the experiment, participants were asked to focus on the ends of phrases that occur throughout the piece--the stimuli was presented only aurally, not visually as in sheet

music. Participants used a PsychoPy-Python program which acts as a stopwatch and returns phrase boundary markings indicated by the user by tapping the spacebar when a phrase ending was detected. The participants were prompted with instructions before beginning the experiment and during the experiment. The program is discussed in greater detail below. The computer that ran the python program (my personal laptop) was sanitized between uses, and I sat at least six feet from the participants as they listened to the excerpts. Both the participant and myself were wearing masks. The participants' data was recorded and imported into an Excel spreadsheet, marked by their anonymous identifier (e.g. Participant 1, Participant 2, etc.). At the conclusion of the experiment, each participant was given a concise verbal overview and formal debriefing statement which provides more detail about the current study and some of the motivating background research. The debriefing statement can be found in Appendix 1. The entire experiment lasted approximately 45 minutes, including the consent process and questionnaire.

PsychoPy Program

The computer program used for this experiment was written using PsychoPy [21]. The program creates a window which displays instructions and details for usage. The space bar was used to mark the participants' perceived, discerned phrase boundaries. The keys "p" and "n" were used for starting and stopping each excerpt, respectively. The pace of the experiment was left to the discretion of the participants, but the order in which the stimuli were presented was always the same. There was a break of indefinite length between the Beethoven and Bruckner stimuli; participants had the option to stand up, stay seated, or simply skip the break altogether.

Once the excerpt starts playing, there was no option to pause--only to skip, if the participant desired to do so. The program has a global stopwatch that resets with the start of each excerpt. The participants' marks for each excerpt were added to a Python list, and that Python list was then exported to a CSV file. The data is being kept secure on an external hard drive.

Section 2.2: Analytical Methods

2.2.1 Praat

For the early stages of exploring audio processing, Praat [13] was a very useful tool to easily load audio files and visually analyze acoustic features. Praat can display waveform analysis and spectral analysis. Praat can also show specific spectral slices, pitch (in terms of frequency), and intensity. Although Praat is more commonly used in speech analysis, it has been a helpful tool to display meaningful audio information and decipher which variables to consider when constructing an algorithm to mark phrases in music.

Praat Text Grids were also juxtaposed onto the display of the waveform and spectrogram. A Praat Text Grid is a text file in which the information written (the length of the audio file, number of intervals, and length of intervals) can show where boundaries are marked in the audio file. When visually juxtaposed onto the audio file, it is much easier to understand what processes are occurring between the markings. The Textgrids were eventually written by hand, but they could also just as easily be written with a Praat script.

Looking at Figure 1 below, from a Praat analysis of the Gregorian Chant stimulus, consider the waveform in the top two panels. It is clear that the amplitude approaches a local minimum near the 14 second mark, as indicated by the taper in the plot. This is what we would

consider to be the end of a musical phrase. The same can be said when considering the spectral analysis of timbre in the middle panel. The vertical, grey bars indicate the richness in overtones at each moment of the sampling rate (44,100 Hz). It is clear that the density of the grey bars decrease as the plot approaches approximately 14 seconds. This visually shows where the phrase appears to be ending.

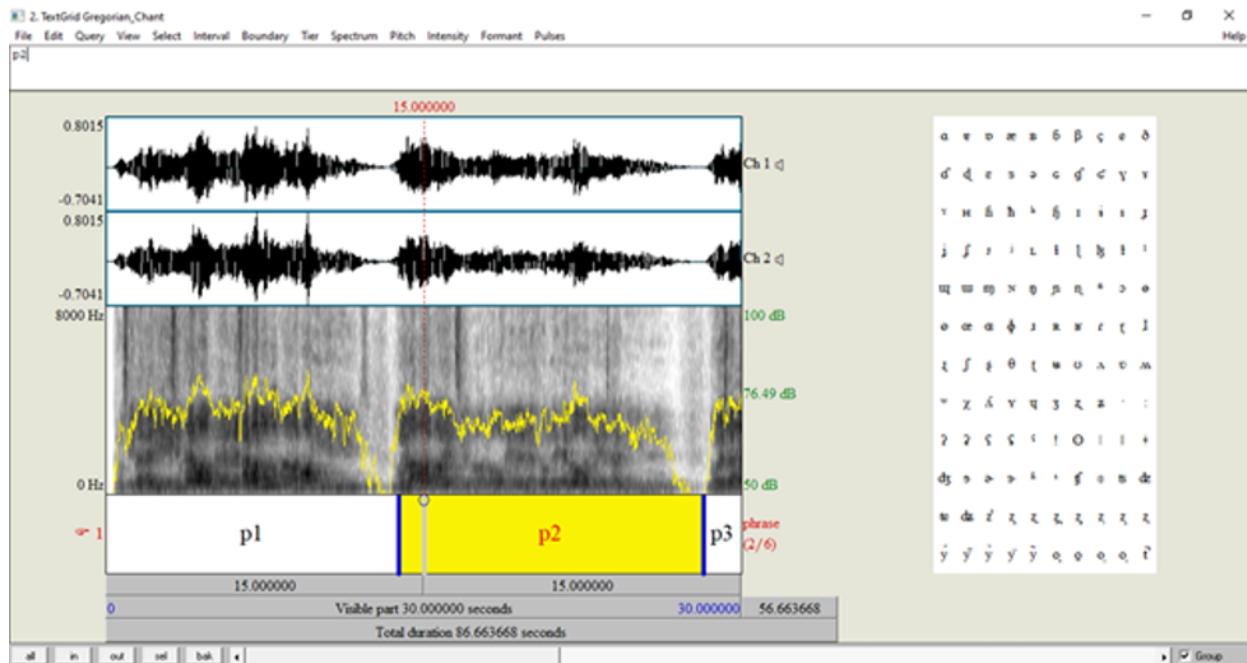


Figure 1: Instance of Gregorian Chant Sample waveform (top two panels), spectrogram (middle panel), intensity (yellow line through middle panel), and Textgrid (bottom panel). The yellow line through the middle panel indicates a fluctuation in contour which denotes the end of a phrase.

2.2.2 Librosa

Librosa is a Python library that handles audio and music processing [14]. The algorithmic approach to finding phrase boundaries in music were handled with Librosa's built-in functions. Audio files are loaded as a Python Numpy array. From the data loaded, we can get the amplitude

at each moment of the sampling rate (44,100 Hz) and represent this as a NumPy array. Using Librosa's unit-conversion functions, we convert the amplitude to decibels, as a NumPy array, and then from decibels to intensity. Librosa also has functionality for acoustic onset detection, and spectral flatness (Wiener Entropy).

The waveform plot for amplitude and spectrogram plot for timbre analyses have been of particular interest when considering variables that determine phrase boundaries. These features are ultimately being used for the development of the phrase-detection algorithm that will loop over a NumPy array and return a set of phrase boundaries from these various constraints.

2.2.3 Acoustic Features

The features extracted from Librosa to use for the detection of phrases in music include intensity, spectral flatness, and rhythmic density (a measure of acoustic onsets per half second). These features were chosen to capture the main musical aspects that constitute a phrase.

Intensity

Loudness, in terms of acoustics, is the amplitude of the sound wave, and intensity is the square of the amplitude. In this way, intensity directly relates to loudness in music. Generally, in Western Classical music, we can gather acoustic information when the loudness fluctuates (Roederer, 2008). When dealing with waveform analysis, intensity is a much cleaner indicator for finding minima. Specifically, the logarithm of the intensity shows quite clearly the troughs of the plot, as shown in Figure 2.

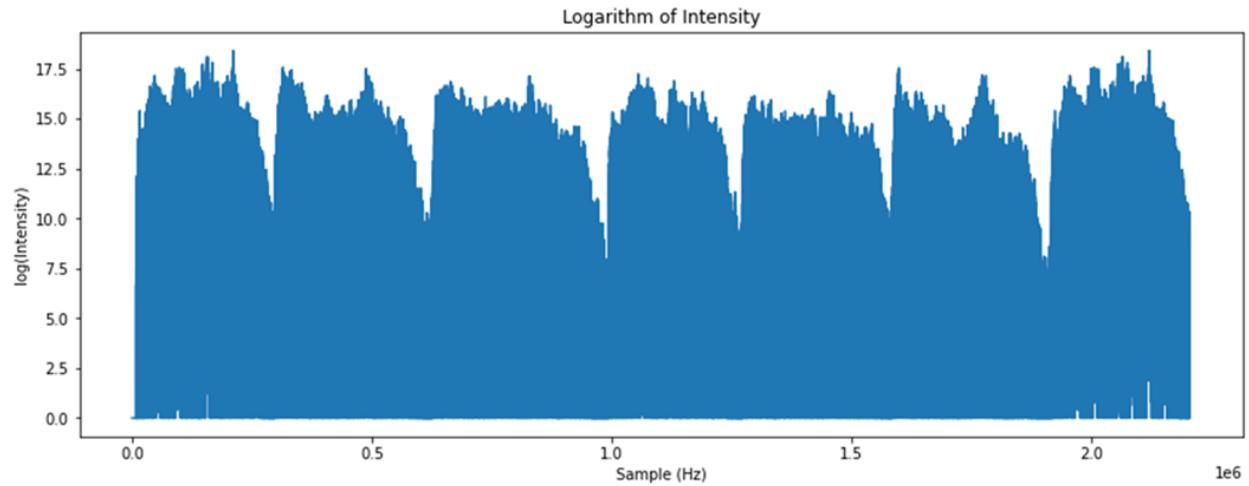


Figure 2: Plot of the logarithm of intensity of the Gregorian Chant Excerpt. Plotted using Librosa. From this plot, there are clear, distinguishable local minima present as the intensity decreases.

Spectral Flatness

Spectral flatness, the tonality coefficient, or Wiener Entropy is used to quantify the audio spectrum over time. Spectral flatness provides a way to describe how tone-like a sound is, rather than white-noise. When the timbre of an audio signal becomes more prominent, the spectral flatness increases proportionally (Olsen, 2016). The spectral flatness is shown as a plot over time in Figure 3, below, from the Beethoven Waldstein excerpt. The spectrogram of this excerpt is shown in Figure 4.

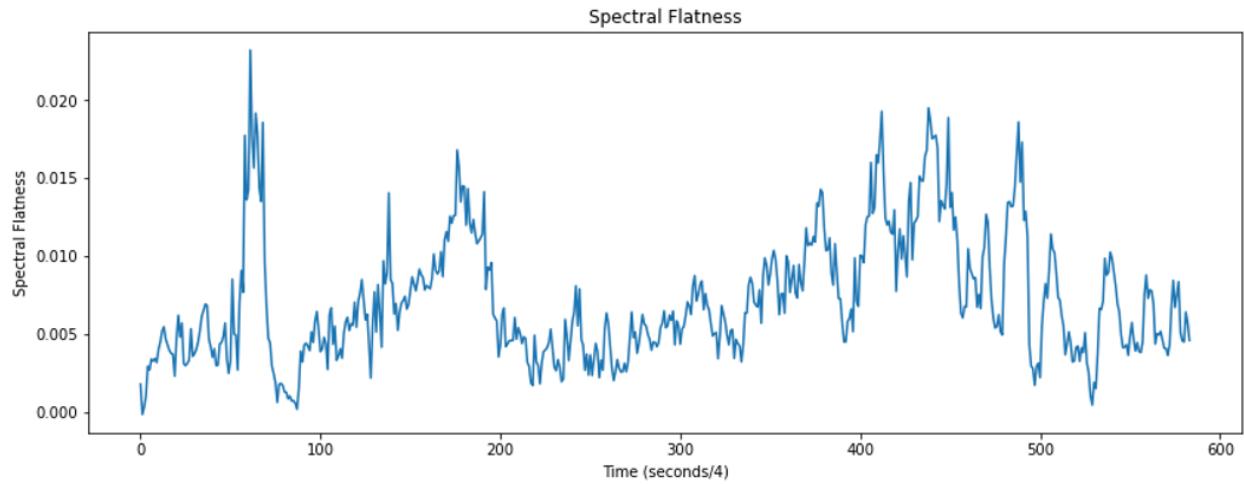


Figure 3: Plot of the spectral flatness of Beethoven's Waldstein. Plotted using Librosa. The x-axis's values are set to four samples per second. There are a few distinguishable local minima that can be seen from the plot, particularly at approximately the 80 mark.

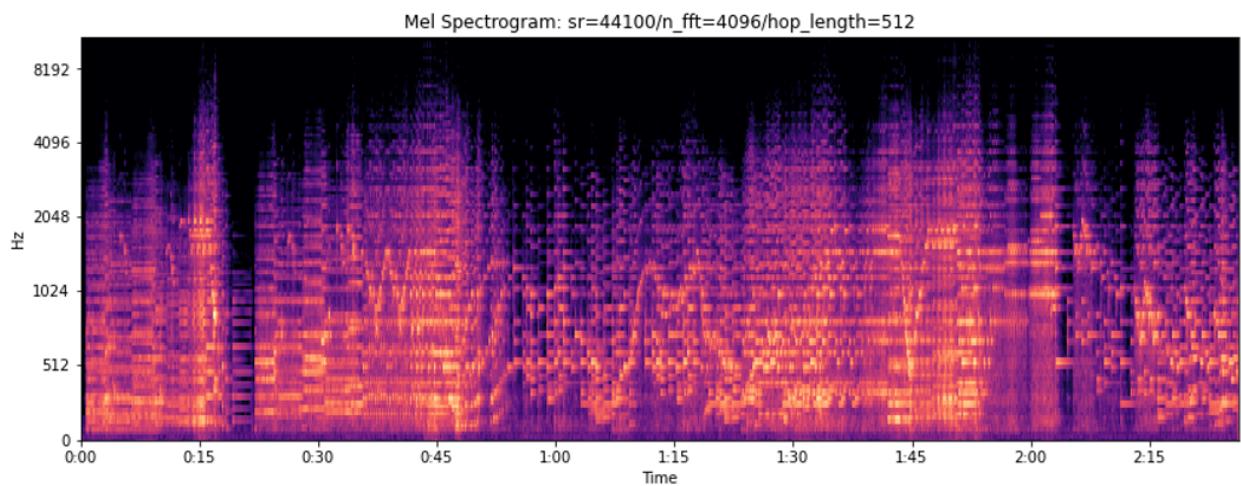


Figure 4: Mel Spectrogram of Beethoven's Waldstein. Plotted using Librosa. The contour of the impressions show the harmonic frequencies present in the sample at each moment in the excerpt. As aforementioned, in Figure 3, there is a clearly distinguishable trough in harmonic frequency at approximately 20 seconds—the 80 mark, when sampled at 4 times per second.

Rhythmic Density

Onsets in acoustics are moments in the audio signal that mark increases in spectral energy, as described in (Degara, 2010). In this study, the threshold for which to record an acoustic onset is set at 8000 Hz, rather than half of the sampling rate (21.5kHz), to have fewer spikes in the signal. The plot of acoustic onsets for the Beethoven Waldstein excerpt is plotted below in Figure 5. Note that the y-axis indicates a proportion of the onset strength, while the red, dashed lines indicate where onsets are most prominent.

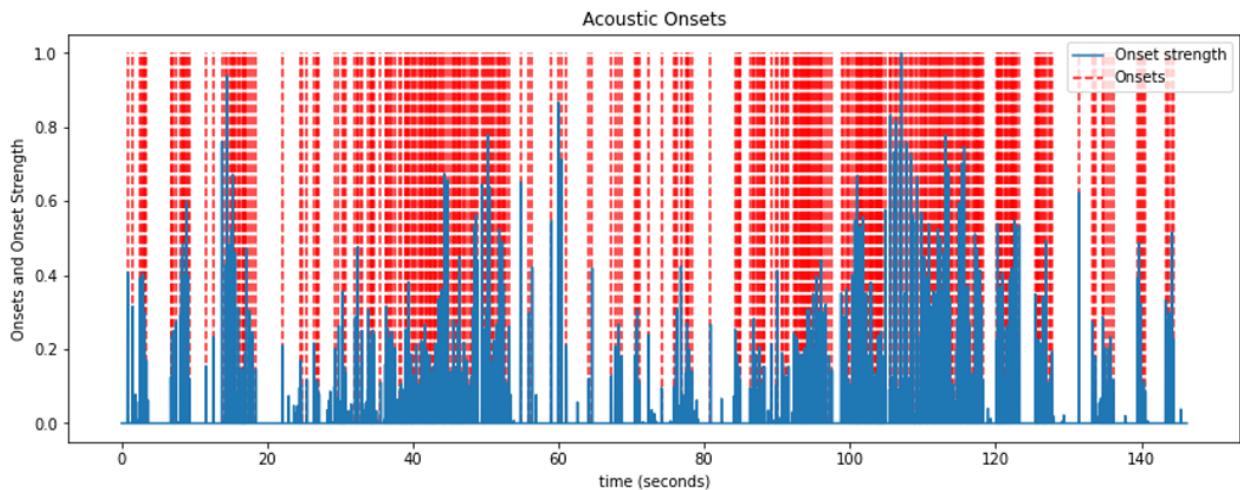


Figure 5: Plot of the acoustic onsets of Beethoven’s Waldstein. Plotted using Librosa. The red, dashed lines indicate where an acoustic onset is perceived. The blue, solid lines show the strength on the onset at each sample.

Rhythmic Density, one of the acoustic features used to detect phrase boundaries, is calculated by summing the number of acoustic onsets every half second, as in the method used by (Olsen, 2016). The Python code for this function, which was implemented specifically for this study, is shown below in Function 1. This particular method was most useful in more rhythmic-driven pieces such as Beethoven’s Waldstein and Mozart’s Symphony No. 40. A plot

of the rhythmic density is shown below in Figure 5. The plot captures the number of acoustic onsets that occur at each half second interval.

```
def rhythmic_density(duration, window, time_steps, onsets_found, ret_array):
    """
    Parameters
    -----
    duration : INT
        Length of the entire audio file
    window : INT
        Specifies the length of window to mark onsets.
    time_steps : ARRAY
        Array that stores the boundaries between each window.
    onsets_found : ARRAY
        Array given that contains onset timings.
    ret_array : ARRAY
        Array, initially empty, that stores the number of onsets found for each window.

    Returns
    -----
    ret_array : ARRAY
        Array that stores the number of onsets found for each window.
    """

    #create initial array that specifies each boundary between windows
    for h in range(1, (duration*2)+1):
        time_steps.append(window*h)

    g = 0 #g is used to increment windows for iteration
    for j in range(len(time_steps)):
        g = time_steps[j] - window #g is always one iteration behind j, to create the window
        g = int(g) #g must be an int to specify range for iteration

        aux_array = [] #auxiliary array used to store each onset within the current window
        for k in range(g, int(time_steps[j])):
            #append k to auxiliary array if the onsets_found array contains k
            if k in onsets_found:
                aux_array.append(k)
            ret_array.append(len(aux_array)) #append number of onsets each iteration

    return ret_array
```

Function 1: Rhythmic density calculation implemented in Python.

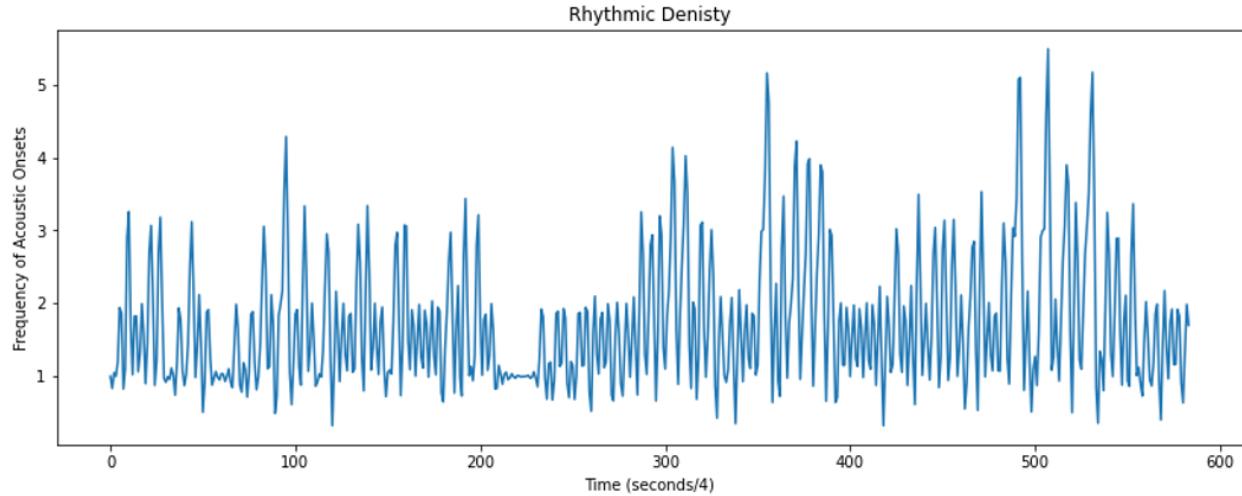


Figure 6: Plot of rhythmic density of Beethoven's Waldstein. Plotted using Matplotlib. The plot captures the number of acoustic onsets found every half second in the signal.

Figure 6 can be compared with Figure 4, the Beethoven Waldstein spectrogram, to see how the frequency of the acoustic onsets correlates to the contour of the spectrogram.

2.2.4 Phrase Detection

Each of the aforementioned acoustic features are objective aspects of the audio signal that are indicative of phrase structure (Olsen, 2016; Knösche, 2005). Although other acoustic features were considered, in each case, the algorithmic approach stays the same--find local minima in a given acoustic feature of the audio signal. Using a function from the Signal module from SciPy [17], `find_peaks`, we can find local minima. This is done by passing the function the inverse of the values of the 1-Dimensional NumPy array of the acoustic features. (the function, `find_peaks`, originally finds maxima in a 1-Dimensional NumPy array, not minima or troughs). A specified threshold indicates the phrase length and tells the function how many seconds to wait before finding the next minimum of the acoustic feature's audio signal. The threshold is necessary to ensure that the function does not mark too many local minima. Even the decimated, resampled

audio signal contains too many minuscule dips in the signal that would be insignificant to a human listener, and thus yield results that don't resemble a human's perception.

Below, Figures 7 and 8 are created by plotting the acoustic features, intensity and spectral flatness of the first excerpt of Bruckner's Symphony No. 7, and using the `find_peaks` function to find minima. In each figure, an orange "X" marks where a significant local minimum occurs. It is the threshold that allows for the algorithm to discern which local minima are chosen over other candidates. In the case of this particular excerpt, the threshold was set at 12.7908 seconds. This threshold was found by taking the average phrase lengths of the participant responses and subtracting 4 seconds from this number, to allow for better recall. Computational analysis of the participants' data was utilized to find the average phrase lengths for each excerpt. The values for each threshold are found by the same method described above: find the average phrase length and subtract 4 seconds. Enhancing the recall, by subtracting 4 seconds, gives the algorithm a better chance of not missing a phrase boundary.

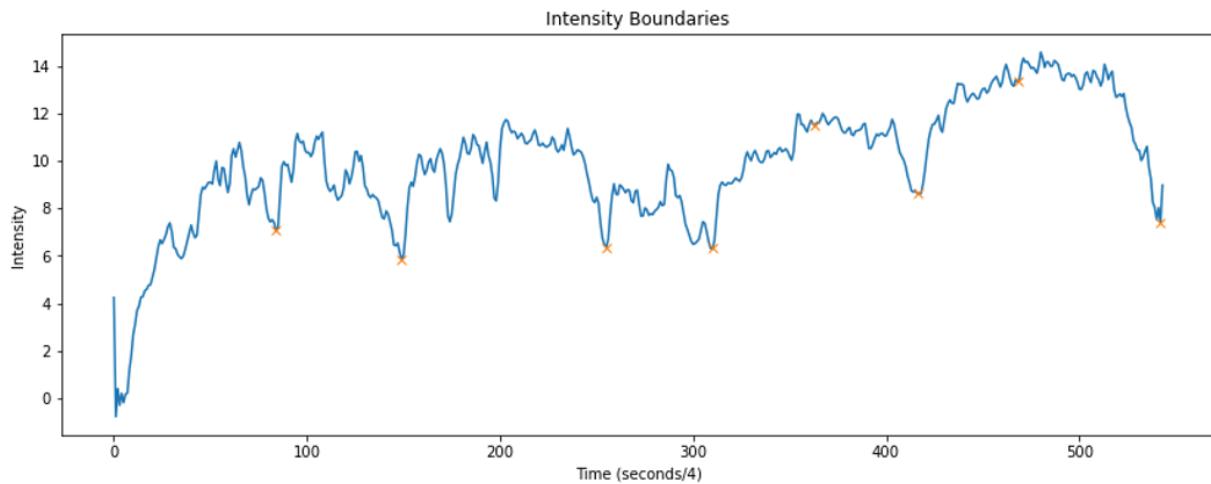


Figure 7: Plot of the intensity of the first excerpt of Bruckner's Symphony No. 7 with local minima shown with orange "X's." The `find_peaks` function from the SciPy module discerned these negative peaks (troughs) from the given threshold.

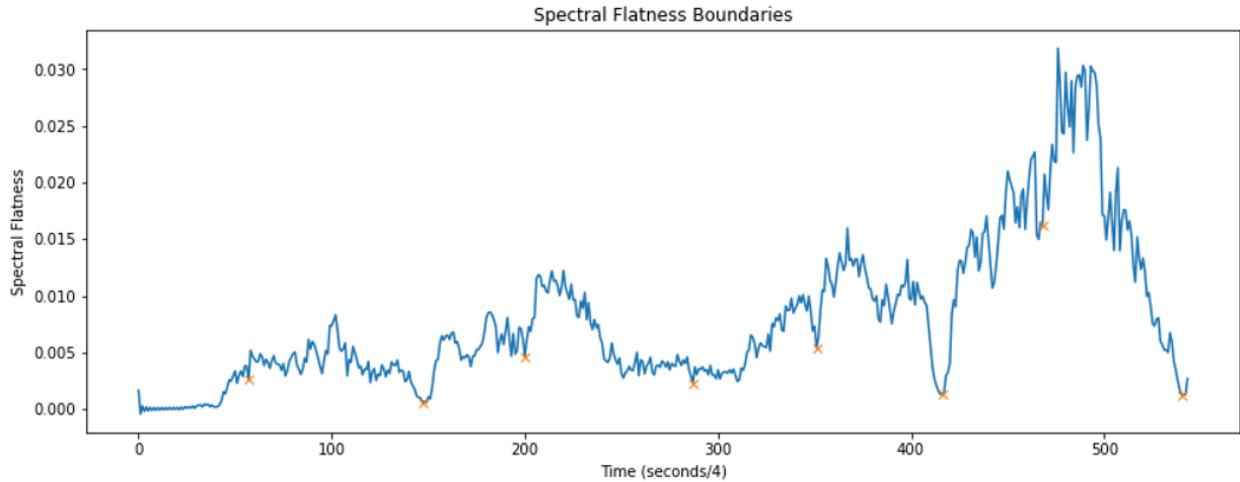


Figure 8: Plot of the spectral flatness of the first excerpt of Bruckner's Symphony No. 7 with local minima shown with orange “X’s.” As with Figure 7, the `find_peaks` function from the SciPy module discerned these troughs from the given threshold.

Section 2.3: Machine Learning

For the machine learning approach to this task, two models were designed and implemented. The first model is a logistic regression model, and the second model is a feedforward neural network. The models take, as inputs, the acoustic feature values at every fourth of a second. The values were extracted from the audio signal using Librosa and were normalized using SciPy [17]. The target function is an array of 0's and 1's. A 1 indicates that a phrase boundary is located at that particular moment in the time frame. Most of the original target array is full of 0's, since the 1's only occur at moments where there is a perceived phrase boundary. It was, however, necessary to widen the window of what is considered to be a phrase boundary. Machine learners cannot give predictions with great accuracy if one of the two classes significantly dominates the other in frequency of occurrence--in our case, too many 0's would lead to the model predicting not enough 1's, or none at all. The window was widened to 4

seconds (2 seconds before the phrase boundary mark and 2 seconds after the phrase boundary mark) for which to indicate a phrase boundary was perceived.

To get the correct mappings for the input array to the target array, a moving window was implemented so that the input value at each fourth of a second would encompass the previous 10 values of each of the acoustic features (a 3 by 10 array--3 acoustic feature values every fourth of a second for 10 fourths of a second). This ensures that the input values (3 by 1 feature vectors) will be associated with the appropriate target values (either 1--there is a phrase boundary, or 0--there isn't a phrase boundary). There are 4020 data points in total--1005 total seconds of each excerpt four times per second. Both models were written in Jupyter Notebooks.

2.3.1 Logistic Regression

Logistic regression is a model that uses a logistic function to classify binary variables. Logistic regression provides a linear decision boundary which predicts the logarithm of the odds ratio so that more independent feature relationships are taken into account (Prabhat, 2017). We can imagine binary classification as a contingency table with two columns for the classes and as many rows as there are data points. We can estimate the log-odds for each row empirically; for many rows, we can interpolate. Logistic regression then applies the logistic function, sigmoid, which is shown below:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

The sigmoid function limits the output to a range between 0 and 1, which is ideal for predicting probabilities. But the sigmoid function is also useful for mapping inputs to binary variables, which is the case in this study (at each sample, there either is a phrase boundary, or there isn't a phrase boundary).

The logistic regression model was designed and implemented in Python using Scikit-learn [15]. The LogisticRegression class was imported from Scikit-learn's linear_model module to implement the model. The training data consisted of 85% of the total data; the testing data consisted of 15% of the total data. This splits the data into 3417 data points for training and 603 data points for testing from 4020 total data points of fourths of seconds for every excerpt. From the LogisticRegression class, the method, fit(), was used to fit the model according to the training data. The model was trained on a maximum of 1000 iterations and ran until the model converged. More specifically, when the model's output got closer and closer to some specific value that ceased to update, the iterative process ended. LogisticRegression's method, predict, was used to get a prediction for the testing data. This prediction is interpreted as the model's guess for the testing data of the target array. The methods called from the LogisticRegression class are shown in Code 1 below.

```
from sklearn.linear_model import LogisticRegression  
  
lg = LogisticRegression(max_iter=1000)  
lg.fit(X_train,y_train.ravel())  
y_pred = lg.predict(x_test)
```

Code 1: Methods called from the LogisticRegression class from Scikit-learn's linear_model module

2.3.2 Feedforward Neural Network

The feedforward neural network was designed and implemented in Python using the Keras module from Tensorflow [16]. When splitting the data for this model, a validation set was used for better facility of training and early stopping. The training data consisted of 80% of the total data, the validation data consisted of 10% of the total data, and the testing data consisted of 10% of the total data. The total data encompassed every data point for just one participant at a time.

The feedforward neural network is a sequential model consisting of four layers: the flattened input layer in a 3 by 1 shape, two fully-connected layers with tanh activations with 3 units each, and a softmax layer with 3 units before the output. The model thus contains 36 parameters for training. Other sizes of models were explored, but the final report of the results is from the size of the model described above.

The tanh (hyperbolic tangent) activation function behaves similarly to the sigmoid function, except this function limits the range of the outputs from -1 to 1, rather than 0 to 1. The formula for hyperbolic tangent is shown below:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

The softmax activation function takes the outputs of the last layer of the network and turns them into probabilities by taking the exponents of each of the outputs and then normalizing each by the sums of those exponents. The sum of the exponents will add up to one. The formula for softmax is shown below:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$

The optimizer used for the model is the Adam (Adaptive Moment Estimation) optimizer.

The Adam optimizer is an optimization algorithm that is used in the place of stochastic gradient descent. The optimization algorithm updates network weights iteratively based on the training data. Adam computes adaptive learning rates for each parameter (Ruder, 2016). The accuracy metric and sparse categorical cross-entropy loss (logarithm loss) metric were used to measure the model's accuracy and loss. The model was fit using early stopping and checkpoints, with 1000 epochs. In the early stopping function, the patience was set at 10 epochs, since the training could, conceivably, not get much better after a few epochs. The design and implementation of the sequential model is shown below in Code 2.

```
import tensorflow as tf
from tensorflow import keras

model = keras.models.Sequential()
model = keras.models.Sequential()
model.add(keras.layers.Flatten(input_shape=[3, 1]))
model.add(keras.layers.Dense(3, activation="tanh"))
model.add(keras.layers.Dense(3, activation="tanh"))
model.add(keras.layers.Dense(3, activation="softmax"))

early_stopping_cb = keras.callbacks.EarlyStopping(patience=10,
                                                 restore_best_weights=True)

checkpoint_cb = keras.callbacks.ModelCheckpoint("best.h5", save_best_only=True)

opt = keras.optimizers.Adam() #learning_rate = 0.001

model.compile(loss="sparse_categorical_crossentropy",
              optimizer=opt,
              metrics=["accuracy"])

history = model.fit(X_train, y_train, epochs=1000,
                     validation_data=(x_val, y_val),
                     callbacks=[checkpoint_cb,early_stopping_cb])
```

Code 2: The design and implementation of the feedforward neural network is a sequential model in Keras, a module of Tensorflow.

3. Results

The results of the experiment and the participant responses are described in this chapter. The participants' data are analyzed to better understand how well humans can detect phrases in music. Summary statistics help to objectively describe and quantify the data. First, to better understand the reliability of the participants, precision, recall, and F1 measures are computed. These metrics are computed to see how well each participant agrees with themselves. Next, frequency and probability diagrams are shown to visualize the collective participant responses for each excerpt. These diagrams show to what extent the participants agree where phrases occur. Then, average phrase lengths and phrases per minute are computed to better understand the duration of the phrases that most musicians perceive. These statistics of the participants' responses set an upper bound for the machine learning approach, since the objective is to mimic human performance.

Section 3.1: Participant Responses

3.1.1 Overview

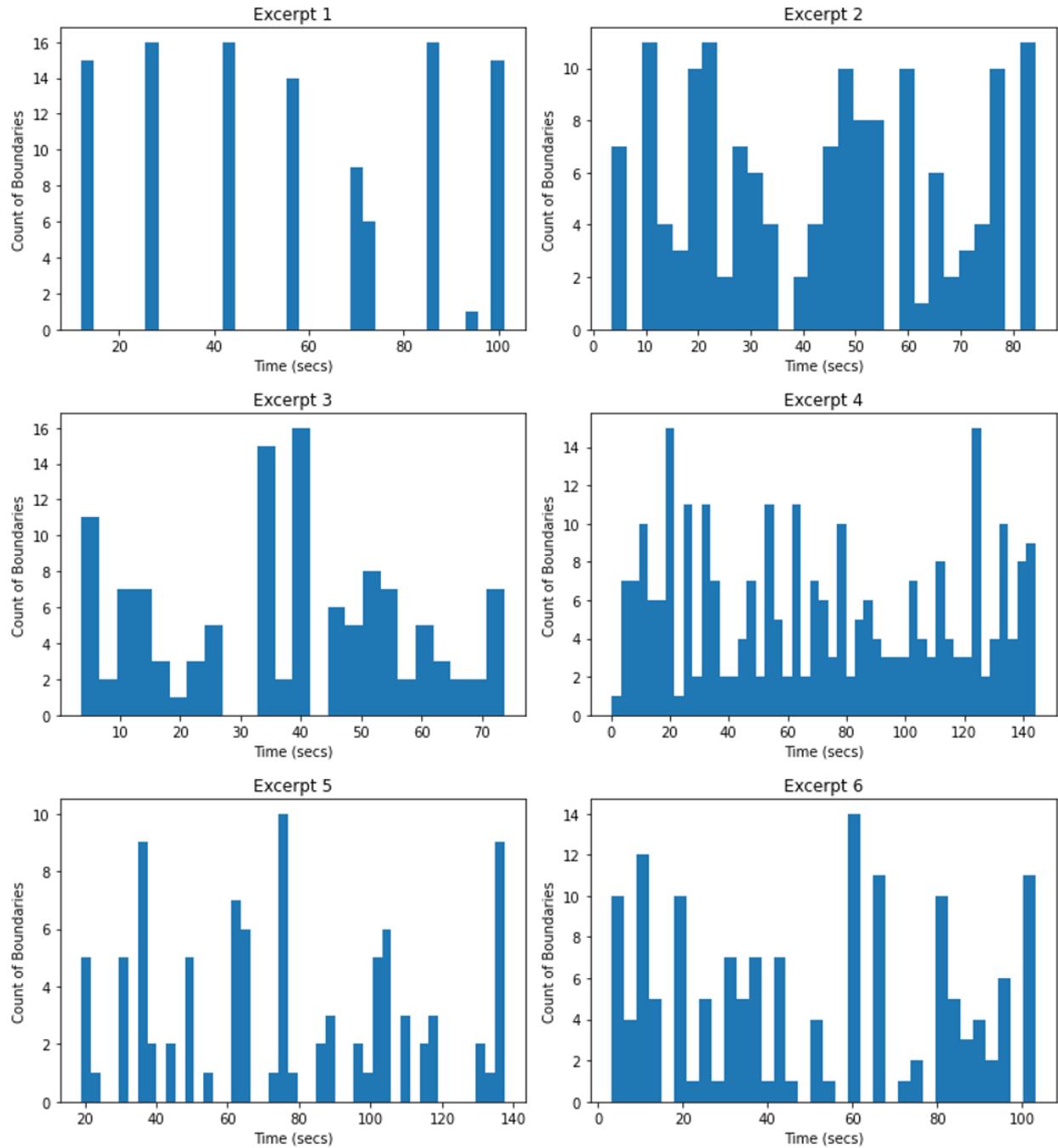
Participants selected for the study had normal hearing and at least three years of formal musical experience. None of the participants noted having any major hearing impairments (e.g., straining when conversing, straining when listening to music, or difficulty when listening to someone's whisper). Three out of the sixteen participants claim to have difficulty discerning nuances in quiet music. Four out of the sixteen participants claim that they often find loud noises cause discomfort. Each participant has taken private music lessons in the past. Thirteen out of the sixteen participants claim that they regularly listen to classical music, as rooted in the traditions of Western culture. Eight out of the sixteen participants have taken either a computer science course or a physics course in the past. Three out of the sixteen participants are familiar with concepts of machine learning. Five out of the sixteen participants are familiar with concepts of acoustics as a branch of physics.

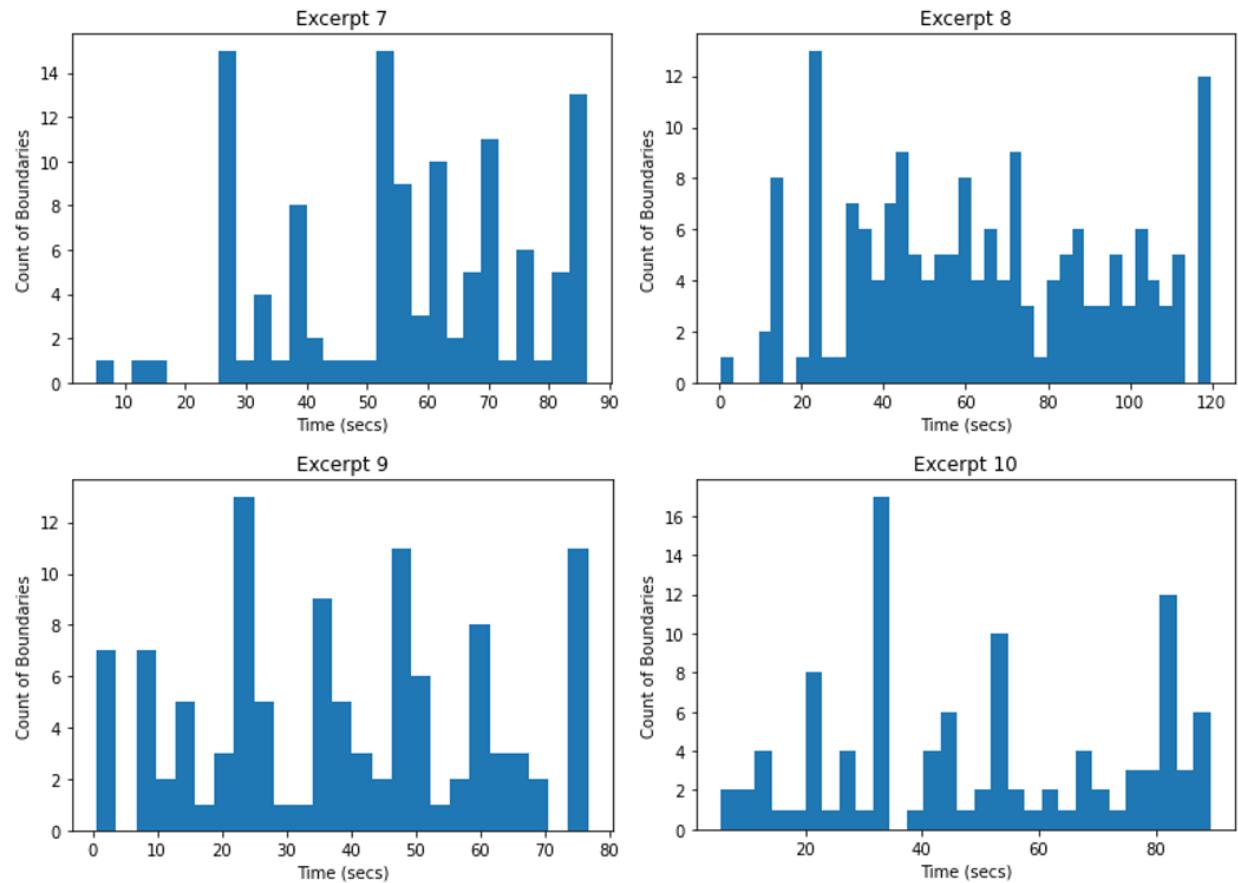
3.1.2 Participant Phrase Markings

The participants' data did not vary significantly, despite a lot of noise in the data. This is contrary to what was expected, since there were plenty of opportunities for human error during the experiment. There were times during the experiment in which participants admitted to "messing up" in their placement of the phrase markings. They perhaps pressed the space bar too early and anticipated what they thought would be a phrase, and realized their mistake after the space bar press was already recorded. This was not an issue, however--there were so many other "correct" or intentional marks from the participants, and the frequency of premature taps was

low in comparison to the intentional marks. The range of disagreement between participants can be seen in the frequency and probability diagrams in Figures 9 and 10. These diagrams show the count of the participants' total number of markings for phrase boundaries for the second listening task of each excerpt. In Figure 9, the histogram bin height indicates the number of times a participant marked a phrase boundary during that particular 3-second window. In Figure 10, the histograms indicate the probability that a phrase will occur within that particular 3-second window.

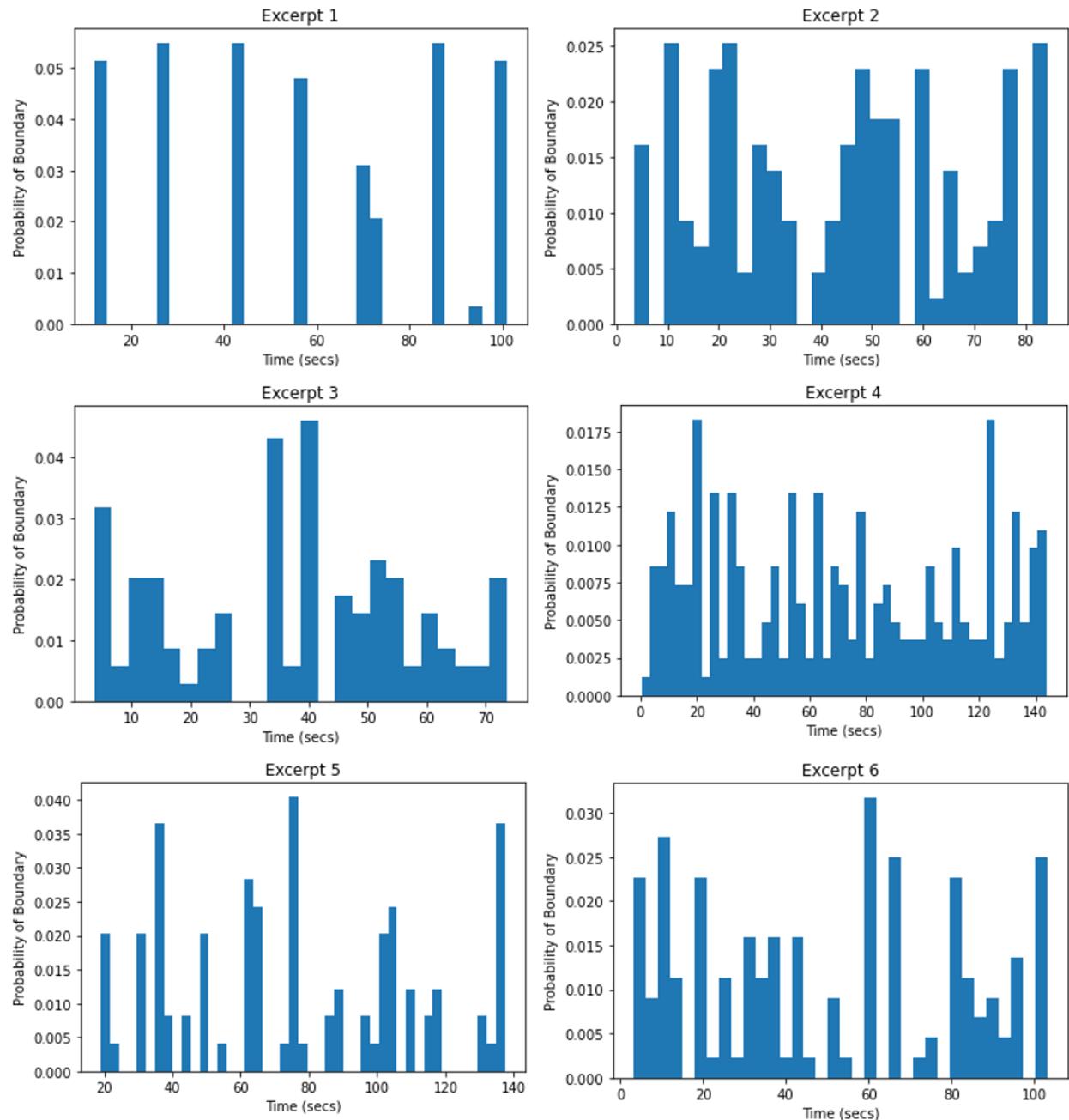
These diagrams show how much agreement is present among the participants. The intra-participant reliability is described in greater detail below in Section 3.2. The noisier histograms illustrate more variation among the participant responses, and less noisy histograms indicate more agreement. The histogram for Excerpt 1 (the Gregorian Chant excerpt), for example, shows much more agreement among the participants. This contrasts with the histogram for Excerpt 4 (the Beethoven Waldstein excerpt) where it is shown that participants disagreed on the durations of phrases and placement of phrase boundaries. Notice, in this histogram, every 3-second window had at least one participant mark a phrase boundary.

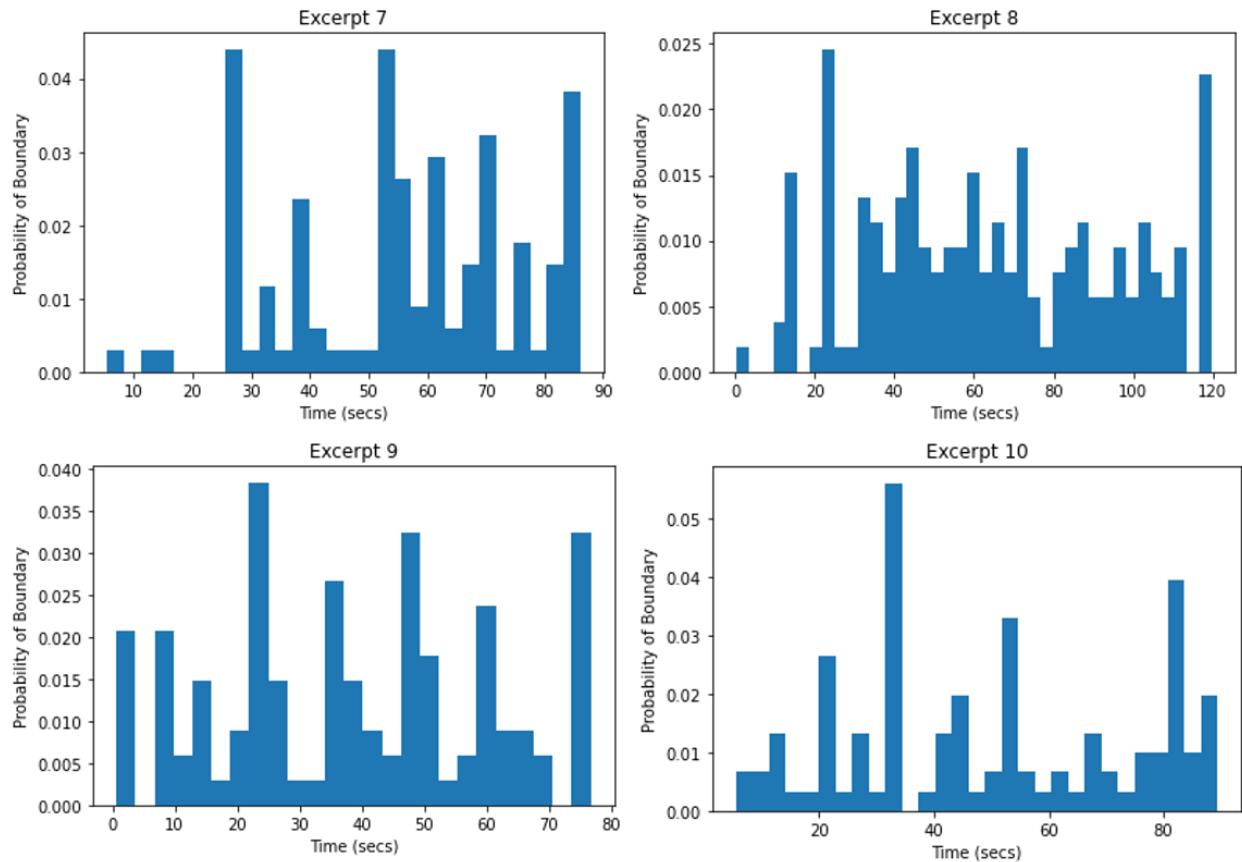




Key: Excerpt 1: Gregorian Chant; Excerpt 2: Mozart exc. 1; Excerpt 3: Mozart exc. 2; Excerpt 4: Beethoven; Excerpt 5: Bruckner exc. 1; Excerpt 6: Bruckner exc. 2; Excerpt 7: Bruckner exc. 3; Excerpt 8: Chopin exc. 1; Excerpt 9: Chopin exc. 2; Excerpt 10: Rachmaninoff.

Figure 9: In each histogram above, the bins indicate how many times the space bar was pressed by each participant. The height of the bins indicate the number of space bar presses from each participant. The width of the bins are in 3-second intervals. The frequency over time tells us how many participants marked a phrase during that particular 3-second interval.





Key: Excerpt 1: Gregorian Chant; Excerpt 2: Mozart exc. 1; Excerpt 3: Mozart exc. 2; Excerpt 4: Beethoven; Excerpt 5: Bruckner exc. 1; Excerpt 6: Bruckner exc. 2; Excerpt 7: Bruckner exc. 3; Excerpt 8: Chopin exc. 1; Excerpt 9: Chopin exc. 2; Excerpt 10: Rachmaninoff.

Figure 10: (Above): The histograms above indicate the probability density of the space bar responses by each participant. The height of each bin indicates the probability that the space bar is pressed from each participant. The width of the bins are in 3-second intervals. The probability density takes the total number of responses into account when producing the probabilities shown on the y-axis. The probability over time tells us the probability that the space bar would be pressed at that particular 3-second interval.

Section 3.2: Participant Reliability

The intra-participant reliability is used to discern to what extent each participant is guessing. In Section 3.4 these intra-participant reliability metrics are used later to determine which participant responses will be taken into account for training the machine learning models.

3.2.1 Precision/Recall Metrics and the F1 Measure

The participants were presented with each excerpt twice during the experiment. The first listen gave the participants an opportunity to become familiar with the style and perceived phrase lengths of each piece. If a participant prematurely tapped the space bar during the first listening task, they were able to rectify their mistake during the second listening task. In addition, the participant's judgements on a single stimulus could be used to measure the reliability of their responses. If musical phrase boundaries can be agreed upon by the participants, then the differences between the first and second listening tasks could determine to what extent the participants were guessing. We don't expect each participant's phrase boundary marks from the two listening tasks to be perfectly aligned. If, however, a participant's two listening tasks produce very similar results, a case can be made that the participant has a firm understanding of where phrase boundaries should be, and that the participant has a concrete idea of what a phrase ending sounds like.

To measure the reliability of each participant, precision, recall and F1 metrics were used. Precision measures the number of correct responses marked over the total number of responses marked (i.e., "out of all x's found, how many are actually x's?"). Recall measures the number of correct responses marked over the total number of correct responses (i.e., "out of all x's out there, how many did you find?"). The F1 metric is used to get a single number as an indicator. The F1 metric calculates the F1-score by the formula shown here:

$$F = \frac{2 \times P \times R}{P + R}$$

To find the precision, recall, and F1-score of each participant, the marks from the second listening task were treated as the “correct” responses. A mark from the first listening task was considered to be found in the second listening task if the two marks were within 3 seconds of each other; there were zero occurrences where the actual value of the marks in the first and second listening tasks matched exactly.

The precision, recall and F1-scores for each participant are shown in Table 1.

Participant	Precision	Recall	F1 Score
1	0.7246	0.7692	0.7463
2	0.8550	0.7870	0.8190
3	0.8846	0.7731	0.8251
4	0.7391	0.7025	0.7203
5	0.8000	0.7477	0.7729
6	0.6400	0.8276	0.7218
7	0.6750	0.6136	0.6429
8	0.7344	0.7833	0.7581
9	0.7800	0.7647	0.7723
10	0.8046	0.8395	0.8092
11	0.7391	0.7907	0.7640
12	0.7532	0.8286	0.7891
13	0.7979	0.6881	0.7389
14	0.7907	0.6296	0.7010
15*	-	-	-
16	0.6833	0.7321	0.7069

Table 1: Precision, recall, and F1-scores for each participant. The precision and recall scores were averaged from each excerpt's individual precision and recall scores by each participant.

*Participant 15 chose to not participate for the first listen.

From the scores shown in Table 1, the mean F1-score is 0.7525, the standard deviation is 0.049401, and the variance is 0.002440. The low standard deviation and variance indicates that the participants did not vary significantly in reliability. Participant 3 recorded the highest

F1-score, 0.8251, Participant 8 recorded the median F1-score, 0.7581, and Participant 7 recorded the lowest F1-score, 0.6429.

3.2.2 Phrase Lengths

The average phrase lengths for each excerpt, as well as average number of phrases per minute, all had slight variation, as was expected. The total average phrase length from every participant, considering every excerpt in this study, was 14.0 seconds, with a total standard deviation of 7.8, and a total variance of 61.0. The average phrase lengths for each excerpt, as well as their standard deviations and variances, separated by each listen, are shown in Table 2. The separation between the first and second listening tasks is shown in Table 2 to compare the differences in perceived phrase lengths between each listening task. It is worth noting that in some of the romantic works (e.g., Bruckner, Chopin, and Rachmaninoff) the standard deviations and variances were generally higher than the works that were written before 1850.

The average number of phrases per minute also is shown in Table 3. These averages were found by taking the total number of phrases found, dividing that number by the total number of seconds of each excerpt divided by 60 seconds, then multiplying that number by the number of participants. This computation is implemented in Python and is shown as Function 2:

```
total_secs = [100, 83, 73, 146, 136, 101, 85, 117, 75, 89] #lengths of each excerpt
phrases_per_min = []
for i in total_secs:
    secs_times_len = (total_secs[i]/60)*len(participants) #len(participants) is 16
    x = count / secs_times_len #count is total num of phrases found
    phrases_per_min.append(x)
```

Function 2: Python implementation of computation used to find total number of phrases per minute.

Excerpt	First Listen			Second Listen		
	Average (secs.)	Std.	Variance	Average (secs.)	Std.	Variance
Gregorian Chant	14.23177	0.103665	0.010746	14.93243	2.767650	7.659886
Mozart exc. 1	9.888642	4.006630	16.05308	10.09660	4.135404	17.10157
Mozart exc. 2	10.07121	3.977049	15.81692	10.48351	4.597006	21.13247
Beethoven	11.48080	4.960965	24.61118	9.858893	3.967461	15.74075
Bruckner exc. 1	16.79077	5.601560	31.37748	20.82815	8.082139	65.32098
Bruckner exc. 2	12.61292	5.588647	31.23297	13.29914	7.328700	53.70984
Bruckner exc. 3	12.80874	5.696789	32.45341	14.94746	8.001132	64.01812
Chopin exc. 1	13.68987	4.196722	17.61248	16.15760	12.94463	167.5635
Chopin exc. 2	14.55226	8.909049	79.37115	13.26375	6.144948	37.76039
Rachmaninoff	24.32566	12.80106	163.8672	17.36317	9.475497	89.78506

Table 2: Phrase length averages, standard deviations of phrase lengths, and variances of phrase lengths for each excerpt, separated by the first and second listening tasks. Since there were only 16 participants, if one participant heard very long phrases, the standard deviation and variance increased dramatically, which explains why these two measures are quite large.

One observation to be made is that as the date in which the piece was composed increases, the standard deviations and variances also increase.

Excerpt	Phrases/Minute
Gregorian Chant	4.7625
Mozart exc. 1	5.7379
Mozart exc. 2	6.5239
Beethoven	3.2619
Bruckner exc. 1	3.5018
Bruckner exc. 2	4.7153
Bruckner exc. 3	5.6029
Chopin exc. 1	4.0705
Chopin exc. 2	6.3500
Rachmaninoff	5.3511

Table 3: Average number of phrases found per minute for each excerpt. These averages were taken from the number of perceived phrases found by each participant, divided by the number of minutes in each excerpt.

Section 3.3: Phrase Detection Algorithm

3.3.1 Acoustic Features as Predictors

My implementation of the `find_peaks` function in the `Signal` module from SciPy found troughs in the signal for each of the acoustic features for each excerpt. These troughs in the signal are interpreted as the Phrase Detection Algorithm's selection of phrase boundaries for every excerpt. Since the algorithm will always mark the same troughs as its phrase boundaries, we did not calculate the F1-score, or any other correlation descriptors, since the result would yield 1.0 each time (i.e., the Phrase Detection Algorithm is based on trough detection and is, therefore, deterministic). The issue of using an arbitrarily chosen threshold to find the troughs is discussed in Chapter 4. The marked phrase boundaries for the Phrase Detection Algorithm for the first Mozart excerpt are shown in Figures 11, 12, and 13 as orange "X's" in the plots.

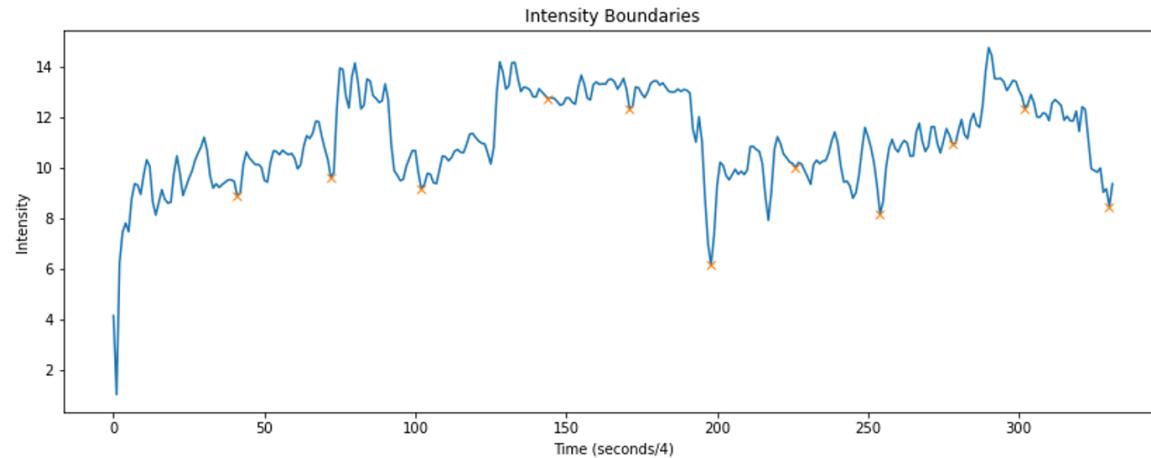


Figure 11: Plot of the intensity of the first excerpt from Mozart's Symphony No. 40 with the Phrase Detection Algorithm's phrase boundaries marked with orange "X's."

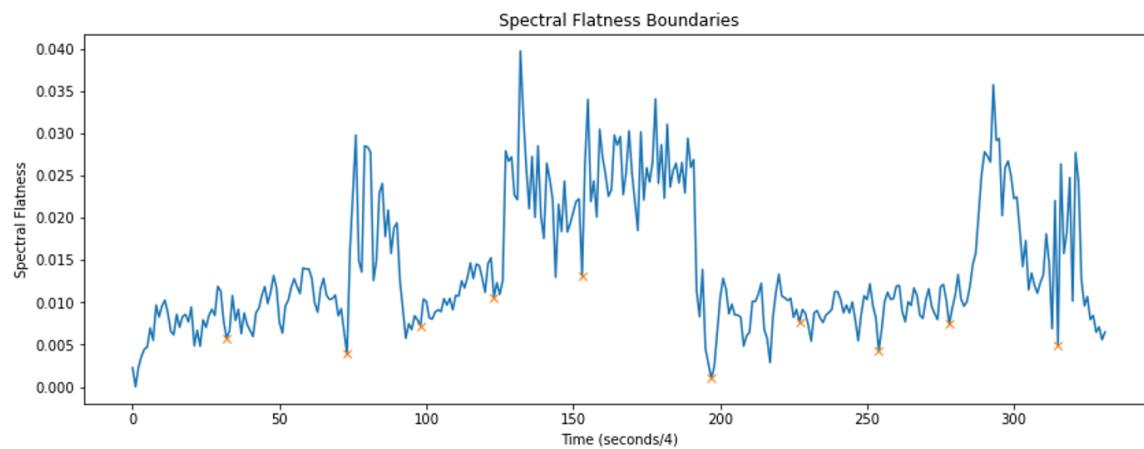


Figure 12: Plot of the spectral flatness of the first excerpt from Mozart's Symphony No. 40 with the Phrase Detection Algorithm's phrase boundaries marked with orange "X's."

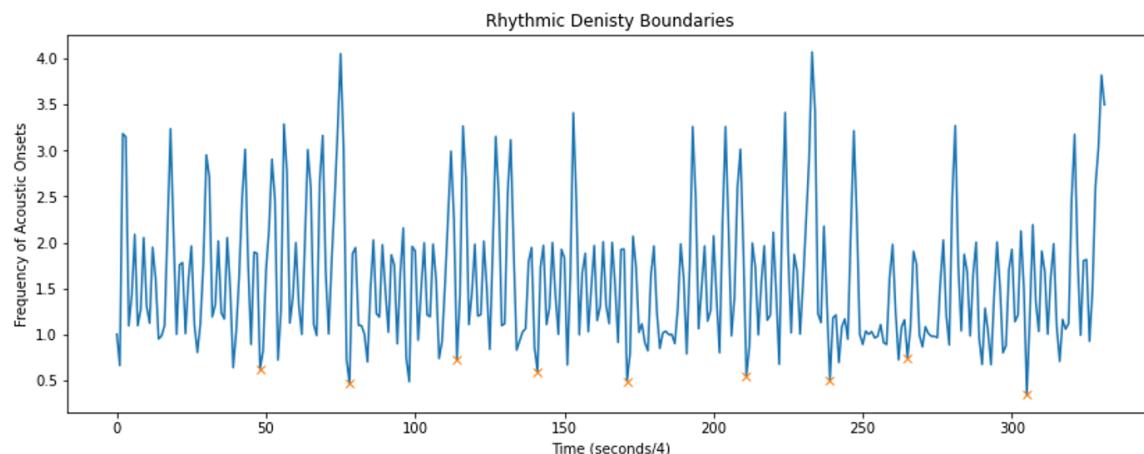


Figure 13: Plot of the rhythmic density of the first excerpt from Mozart's Symphony No. 40 with the Phrase Detection Algorithm's phrase boundaries marked with orange "X's."

All of the plots for the Phrase Detection Algorithm's output for each of the acoustic features can be found in Appendix 2.

3.3.2 Acoustic Features compared with Participants

The variances of the average phrase durations of the Phrase Detection Algorithm's output can be compared with the average phrase durations of the participant responses by using a Statistical F-Test. The Statistical F-Test is computed by dividing the variances of the average phrase durations for the Algorithm by the variances of the average phrase durations for the participant responses. The formula is shown below:

$$F = \frac{s_1}{s_2}$$

where s_1 and s_2 are the variances of the average phrase duration for each of the excerpts, for each of the outputs.

In Table 4, the F-scores for the average phrase durations of each of the acoustic features compared with the average phrase durations of each of the participant responses are shown. In each case, a higher F-score indicates that the correlation between the acoustic feature and the participant response was stronger.

Excerpt	Participants v. Intensity	Participants v. Spectral Flatness	Participants v. Rhythmic Density
Gregorian Chant	109.0021	107.4587	107.2318
Mozart exc. 1	31.06460	32.21992	29.09205
Mozart exc. 2	18.37395	19.24093	19.71520
Beethoven	97.56608	118.7535	103.1564
Bruckner exc. 1	19.90060	21.85093	22.71489
Bruckner exc. 2	11.29601	13.38277	14.71152
Bruckner exc. 3	8.594223	9.639227	9.100488
Chopin exc. 1	6.083678	6.486119	2.685402
Chopin exc. 2	12.05362	11.45003	9.383595
Rachmaninoff	5.402177	7.955517	7.181322

Table 4: F-scores to compare variances between the average phrase duration of the participant responses and the average phrase duration of the Phrase Detection Algorithm's output for each acoustic feature. The higher F-score indicates a stronger correlation between the participant responses and the Phrase Detection Algorithm's output for each acoustic feature.

Section 3.4: Machine Learning Models

Both of the models were trained and tested on just three participants: the participant with the highest F1 score of reliability, the participant with the median F1 score of reliability, and the participant with the lowest F1 score of reliability. Using participants with varying F1 scores yielded quite different results of the metrics used to measure the success of the model. But the difference in how many phrase boundaries each of these participants marked was a critical factor in producing the results herein. Recall that the target function for the model is an array of 0's with batches of 1's, encompassing a 4-second window, whenever a phrase boundary is perceived. If a particular participant heard many phrases and marked a plethora of phrase boundaries (as in Participant 3, used for these results), then the target function actually consists of more 1's than 0's. The numbers of phrase boundaries that each of the other two participants

used to get these results were much lower (Participant 8: 60 boundaries marked; Participant 7: 48 boundaries marked) than the number of boundaries marked by Participant 3 (129 boundaries marked).

3.4.1 Logistic Regression Model

The logistic regression model was scored using accuracy, logarithm loss, precision, recall, and F1. The accuracy metric is used to measure the number of correct predictions made for detecting a phrase boundary (correct 1's in the prediction array), over the total number of actual phrase boundaries marked by the participant (all 1's in the target array). The logarithm loss metric is used to quantify the accuracy metric by penalizing incorrect classifications (i.e. predicting a phrase boundary where there is no boundary, or neglecting to predict a phrase boundary where there is a boundary present). The logarithm loss function can be defined as follows:

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log p_{ij}$$

N is the number of samples (every fourth of a second), M is number of labels or classes (since this is binary classification, M is 2), y is an binary indicator of whether or not label j is the correct classification for sample i , and p is the probability of assigning this label j to sample i .

For each participant, the accuracy and log loss of the logistic regression model was calculated and is shown in Table 5. From the table, Participant 3's accuracy score is much higher

than Participant 8's and Participant 7's accuracy scores. As aforementioned, Participant 3 had marked many more phrase boundaries than Participant 8 or Participant 7. This high number of 1's in the target array led to the model leaning toward mostly predicting boundaries. This issue is addressed below when computing the specificity.

Accuracy and Log Loss for Participants 3, 8, and 7	
Participant 3	
Accuracy	0.9834
Log Loss	0.5728
Participant 8	
Accuracy	0.7944
Log Loss	7.1027
Participant 7	
Accuracy	0.7214
Log Loss	9.6230

Table 5: The accuracy and log loss shown from running the model on each of the above participants' target arrays. Participant 3's accuracy score is much higher than Participant 8's and Participant 7's accuracy scores since Participant 3 had marked many more phrase boundaries than Participant 8 or 7. This led to the model leaning toward mostly predicting boundaries.

These varying accuracy and log loss scores indicate that it is particularly difficult for the model to accomplish the task of predicting phrase boundaries without overfitting or underfitting, given the training set size of 3417 and the testing set size of 603. Had the window of 1's allotted for a single phrase boundary been any smaller, the model would have done mediocre on the target array for Participant 3 (since Participant 3 recorded many more phrase boundaries) but quite poorly on the target arrays for Participants 8 and 7.

The probabilities for each of the target value predictions are modeled in Figures 14, 15 and 16. The blue data points are the target array's values at its appropriate input value.

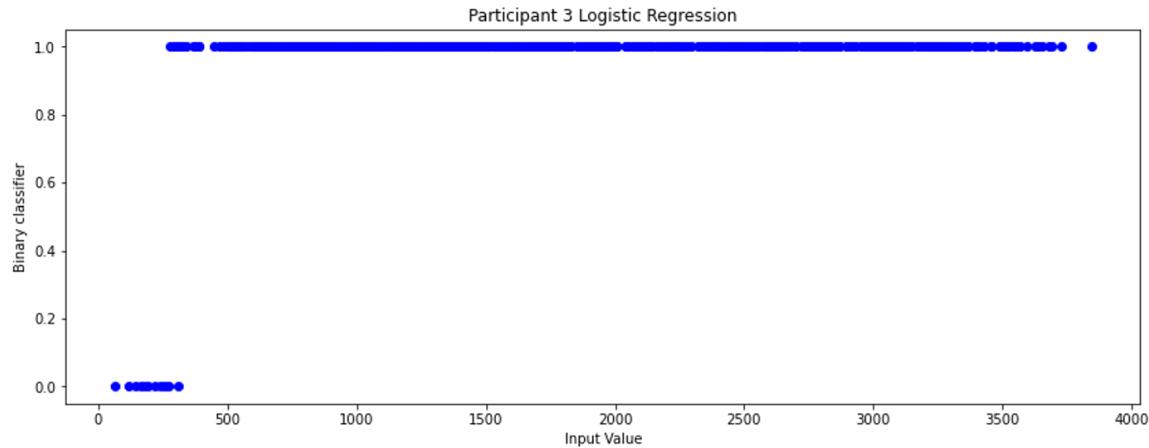


Figure 14: Plot of each value in the prediction of Participant 3's target array by running the logistic regression model.

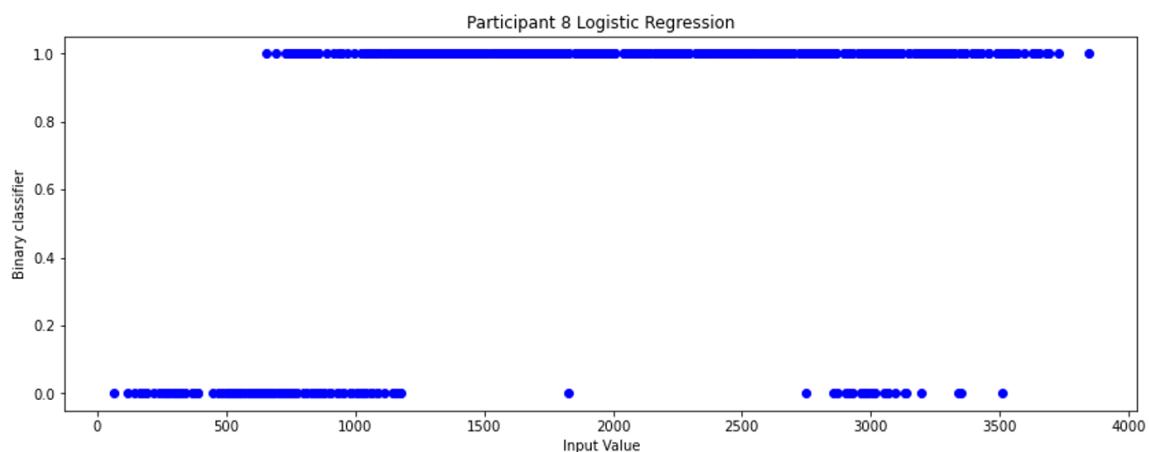


Figure 15: Plot of each value in the prediction of Participant 8's target array by running the logistic regression model.

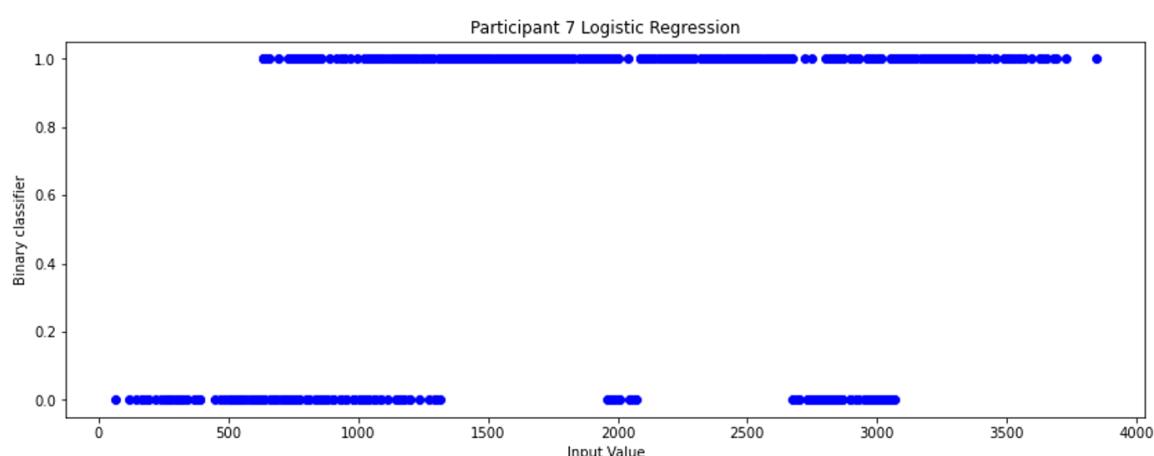


Figure 16: Plot of each value in the prediction of Participant 7's target array by running the logistic regression model.

A confusion matrix provides a concise description of predicted versus actual classifications to describe the performance of the model for each run on the participant's target functions (Prabhat, 2017). A confusion matrix indicates the number of true negatives (top-left), false negatives (bottom-left), false positives (top-right), and true positives (bottom-right). True negatives are predicted 0 and are actually 0, false negatives are predicted 0 but are actually 1, false positives are predicted 1 but are actually 0, and true positives are predicted 1 and are actually 1. Table 6 shows the confusion matrices for the runs of the model for each participant.

Confusion Matrices for Participants 3, 8, and 7			
Participant 3			
n = 603	Predicted 0	Predicted 1	Actual Totals
Actual 0	4	10	14
Actual 1	0	589	589
Predicted Totals	4	599	
Participant 8			
n = 603	Predicted 0	Predicted 1	Actual Totals
Actual 0	5	124	129
Actual 1	0	474	474
Predicted Totals	5	598	
Participant 7			
n = 603	Predicted 0	Predicted 1	Actual Totals
Actual 0	4	166	170
Actual 1	2	431	433
Predicted Totals	6	597	

Table 6: Confusion matrices from running the logistic regression model on Participants 3, 8, and 7. The totals for actual and predicted classifications are shown at the extremities of each row and column of each confusion matrix.

The precision, recall, and F1 metrics provide quantifiable results from the confusion matrices above. Previously, in Section 3.2.1, precision, recall and F1 scores were used to measure the intra participant reliability. The use of those metrics led to the selection of Participants 3, 8, and 7 to train and test the model because their F1 scores were the maximum, median, and minimum, respectively, among all of the participants. But precision, recall, and F1 metrics are more classically used to score models that predict classifications. The scores for each of these metrics can be found using the confusion matrix. Precision is found by dividing the number of true positives by the total number of predicted 1's. Recall is found by dividing the number of true positives by the total number of actual 1's. The F1 metric is found by multiplying 2 by precision and recall and dividing by the sum of precision and recall. This formula is shown again, below.

$$F = \frac{2 \times P \times R}{P + R}$$

The F1 metric gives a single-number estimator to the precision and recall metrics.

Sensitivity and specificity (true positive rate and true negative rate, respectively) gives a score for the recall and precision, respectively. Recall and sensitivity are computed in the same way. Specificity reports the precision on the negation of the task at hand. This measure indicates how well the model performs at finding data you don't want.

The precision, recall, F1, sensitivity, and specificity scores for the model's runs of each of the participant target arrays are shown in Table 7. These scores for the model's run on Participant 3's target array are quite high, for the same reasons described above in discussing the

accuracy and log loss scores. The same issue arises here; the model tends to overfit or underfit depending on the size of the phrase boundary window. This issue is discussed in greater detail in Chapter 4. For these target arrays, the model did very well with Recall--the model's prediction only missed two phrase boundaries, in total.

Precision, Recall, F1, Sensitivity, Specificity for Participants 3, 8, and 7					
Participant	Precision	Recall	F1	Sensitivity	Specificity
3	0.9833	1.0	0.9916	1.0	0.2857
8	0.7926	1.0	0.8843	1.0	0.0388
7	0.7219	0.9954	0.8369	0.9954	0.0235

Table 7: Precision, recall, F1, sensitivity (true positive rate), and specificity (true negative rate) scores for the logistic regression model's runs of each participant's target array. The recall and sensitivity for the target arrays for Participants 3 and 8 are 1.0 because every 1 in the target array was found by the model's prediction.

The issue described above about the model mostly predicting boundaries is shown through the low specificity. The model does not do well discerning true negatives in any of the cases.

3.4.2 Feedforward Neural Network Model

The feedforward neural network model was run for the same three participants' target arrays. The input arrays and target arrays are unchanged between this model and logistic regression model. The same notion arises here, that the difference in how many phrase boundaries each of these participants marked has a significant impact on the accuracy and log loss scores. In Table 8, the accuracy and the log loss scores are shown before and after the model was trained.

Test Accuracy/Log Loss Scores Before and After Training				
Participant	Initial Accuracy	Initial Log Loss	Accuracy After Training	Log Loss After Training
3	0.0822	1.9144	0.9151	0.2931
8	0.3263	1.2635	0.6710	1.7806
7	0.5913	0.9138	0.5969	1.8850

Table 8: The accuracy and log loss scores on the testing data before and after training the sequential model. The target arrays for each of the three participants were individually trained and tested to yield these results.

The validation set is taken as a subset of the training data. In training the sequential model, the validation set was used to tune the model's parameters to avoid overfitting. The validation accuracy and log loss scores for each target array for this model are shown below in Table 9. These scores support the notion that generalization was quite good, but the scores for log loss do not align closely with the scores for log loss in the testing data, perhaps contradicting the notion that generalization was good. High log loss indicates that the model had many incorrect classifications, while still predicting most of the phrase boundaries that were present in the target arrays.

Validation Accuracy/Log Loss Scores After Training		
Participant	Validation Accuracy	Validation Log Loss
3	0.9333	0.2228
8	0.9490	0.1208
7	0.9294	0.1719

Table 9: The accuracy and log loss scores on the validation set after training the model. The target arrays for each of the three participants were individually trained and tested to yield these results.

The next three figures, Figures 17, 18, and 19, each show the progression of training on a maximum of 1000 epochs for each of the three participants' target arrays. The training on Participant 3's target function stopped at 606 epochs since it didn't appear to be improving over the previous 10 epochs.

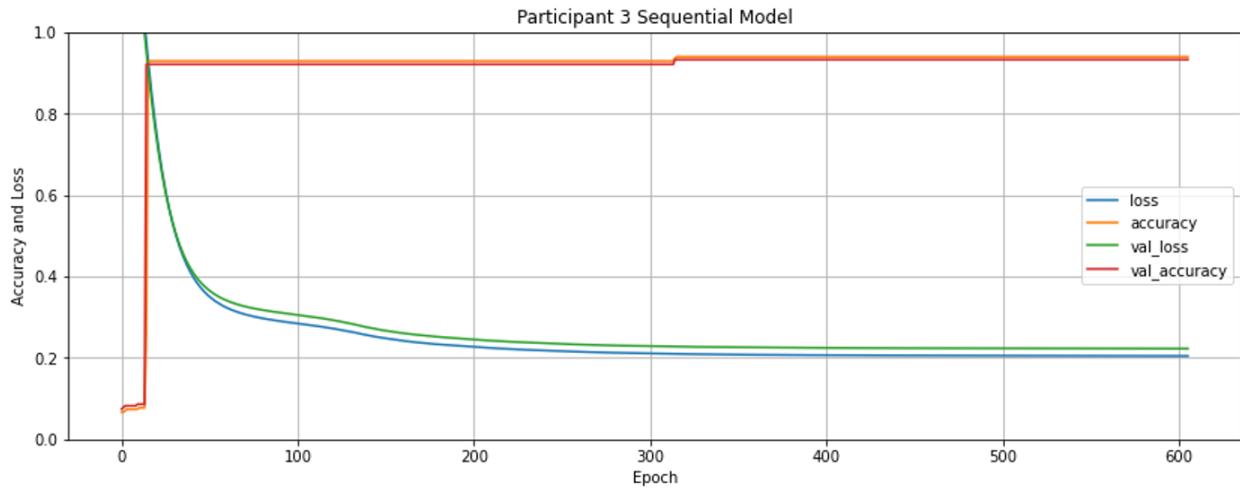


Figure 17: The progression of training for the sequential feedforward neural network on Participant 3's target array.

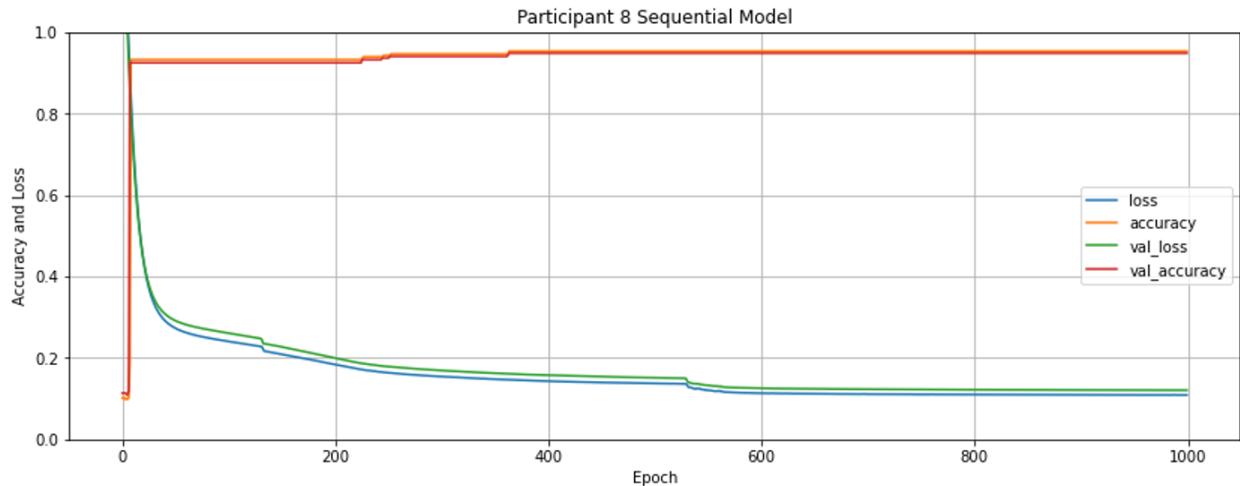


Figure 18: The progression of training for the sequential feedforward neural network on Participant 8's target array.

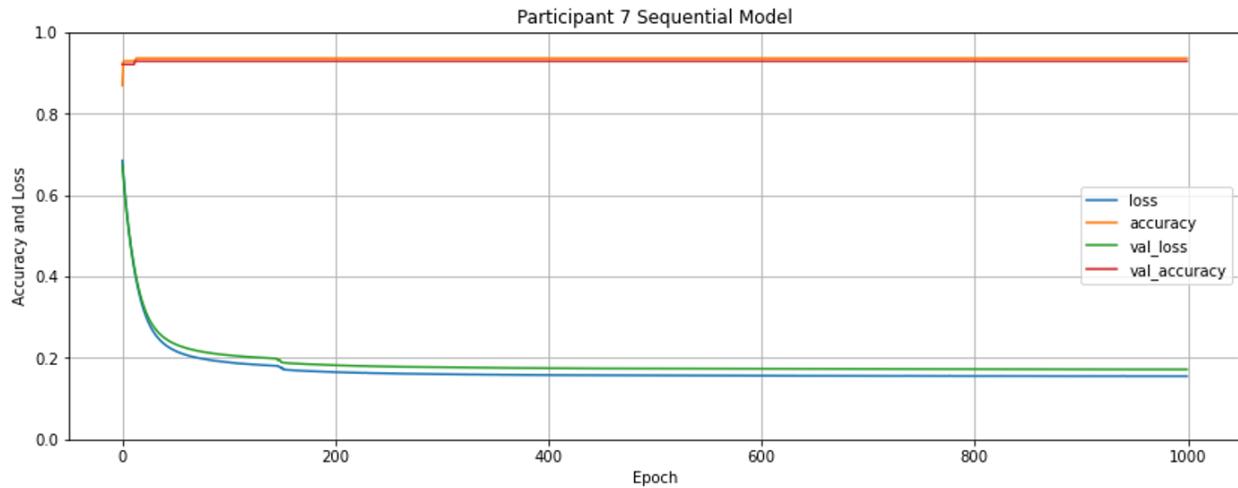


Figure 19: The progression of training for the sequential feedforward neural network on Participant 7's target array.

4. Discussion

This project had multiple objectives. The project aimed to define phrase boundaries in music of the Western classical tradition, and to create a working, efficient algorithm that can detect phrase boundaries in music. These objectives were achieved in many different ways. The original ideas for the algorithm started as broad concepts, then narrowed into more specific problems such as feature extraction and binary classification. The project ultimately accomplished the goals of defining phrase boundaries and creating an algorithm, but there are particular areas of the project that are left for the next stage of research. In this chapter, a few of these areas are outlined, with suggestions on how to rectify and improve them moving forward.

Analysis

The results implicate that the Phrase Detection Algorithm and the two machine learning models perform well for the task of predicting phrases in music, using the participant responses as the standard for correctness. The results of the two approaches, the algorithmic and machine

learning approaches, are different in how they predict. The Phrase Detection Algorithm predicts a single mark, whereas the machine learning models predict phrase boundary windows, but in either case, the results strongly resemble the participant responses, with low variance, in the case of the Algorithm, and high accuracy, in the case of the machine learning models.

Section 4.1: Evaluation

The processes of conducting the experiment, analyzing the data, and preparing the data for machine learning models produced meaningful results. The processes were not perfect, and there are many ways that each aspect of this study could be improved. Outlined below are objectives in the study that could have been conducted differently, with suggestions for improvement.

4.1.1 Experiment

The experiment was designed with the goal of gathering data for a machine learning algorithm and ensuring the safety of the participants in the process. Ultimately, the experiment succeeded in accomplishing these objectives, but there are still amendments to be made to the experimental procedures for any researcher wishing to replicate the experiment. In particular, the experimental procedure should be straightforward, and the communication between the researcher and the participants should be explicit and unambiguous. The adherence to these practices is the intention, but the execution of these practices could always be better.

Design of the Program

The purpose of the experiment was to gather data from musicians to use as training and testing data for a machine learning algorithm. The experimental procedure was consistent for each participant, although some participants had difficulty using the program. This is ultimately a design flaw--the program should have been designed to be extraordinarily simple, so that anyone, at any level of knowledge about computer programming or technology, could use the program as it was intended. The downside of assuming any prior knowledge leads to unexpected results in unexpected ways. In this case, a design flaw which required the participant to press two keys between stimuli, led to losing data on the first stimulus after the break. Since the break was treated as another stimulus by the program, the process of pressing “n” for stop and “p” for start also applied here. From the perspective of the participant, at the end of a break, the natural way to continue the task would be to press start first, not stop. The loss of this data was ultimately unimpactful with the exception of affecting some the participants’ F1 scores when computing intra-participant reliability. In the case of Participant 15, there was simply a communication issue with the experimental procedure. Participant 15 used the first listen as an opportunity to become familiar with the style of each excerpt, but didn’t realize the task was to mark phrase boundaries on both the first and second listening tasks. Because of this, Participant 15’s data was not vetted with F1 scores and thus not considered when choosing participant data for the machine learning algorithm.

Stimuli Biases

The stimuli chosen for this experiment were all from pieces of the Western Classical tradition. Some of the pieces were familiar to most of the participants, while other pieces were

unknown to most of the participants. The variability in familiarity was desired to allow the participants to become familiar with the listening task on pieces that they recognized, then introduce a few pieces that most of the participants had heard before so that any prior knowledge of the piece did not affect the responses. This practice of using pieces familiar to the participants, however, introduces unintentional biases. For the machine learner, biases can influence the ability to generalize or make predictions on new material. Having chosen a few pieces that were unfamiliar to the participants, the hope was that these biases were reduced; although these biases can never be fully eliminated. It is advised in the future, if this experiment is replicated, that the pieces chosen be pieces that are likely to be unknown to the participants. It is perhaps an intuition that trained musicians will have no difficulty understanding phrase structure and thus detecting phrase boundaries in unfamiliar music.

4.1.2 Algorithmic Evaluation

The Phrase Detection Algorithm added many useful insights to this study. The Algorithm affirmed an intuitive understanding of where phrase boundaries occur, relative to acoustic cues in the signal. There are a few ways that the use of this algorithm could further improve the results of this study.

Additional Acoustic Features

For this study, only three acoustic features were used in the detection of phrase boundaries. These three features (intensity, spectral flatness, and rhythmic density) were chosen because they each represent a quantifiable measure of fluctuations in the sound that relay

information (Olsen, 2016; Knösche, 2005; Roederer, 2008). In (Olsen, 2016), these three features were central to discovering what constitutes a phrase for sound-based music, and in (Knösche, 2005), these acoustic features were found to fluctuate similarly in speech and in music. It is through these previous studies that the justification arose for using only these three features for this study. In future implementations of this work, more acoustic features like zero-crossing rate (a measure of the number of times the value of the signal crosses the zero axis), equivalent rectangular bandwidth (the value of which is a function of center frequency), spectral crest (obtained by comparing the maximum value and arithmetical mean of the spectrum), and other candidate features should be considered (Peeters, 2011).

Arbitrary Thresholds

The use of a threshold value for the Phrase Detection Algorithm was problematic, considering how it was arbitrarily computed. Recall that the threshold was found by taking the average phrase lengths of the participant responses and subtracting 4 seconds from this number. A better approach would have been to smooth the signal even more than the process of resampling was able to do. Smoothing the signal would have perhaps eliminated many of the unimportant troughs in the signal, allowing for the `find_peaks` function to easily detect the meaningful troughs on its own.

4.1.3 Machine Learning Evaluation

The machine learning approach ultimately delivered interesting results, but there are a few areas that could have been handled differently. One example discussed here is the size of the

windows for the target arrays; the window sizes should have fluctuated with the duration of phrases for each participant. Other aspects of the machine learning approach fell short of getting better results, but this was ultimately due to having so little data available for training more sophisticated models for the task.

Target Arrays

When creating the target arrays for the machine learning algorithm, the small window of each phrase boundary mark (just one 1 every mark) proved to be an issue when training the model. In real time, the mark is minuscule, so it could conceptually be just a single 1; but, to the machine learner, which treats the problem as a classification task, the mark would be difficult to discern if it were represented by a few single 1's in a vast array of 0's. For logistic regression models or for small sequential neural networks, unequal representation of a class causes the classification predictions to be strongly skewed to the dominant class, and less sensitivity to features that predict underrepresented classes.

To overcome this issue, the window for a phrase boundary mark was widened, thus giving the algorithm a chance to detect a different class within the window. In this study, the window was widened from 1 second to 4 seconds (2 seconds before the mark, and 2 seconds after the time indicated by the listener). Widening the window provided a solution to the classification problem, and allowed the machine learner to make accurate predictions on the testing data. An improvement to this solution would be to vary the size of the window, proportional to the number of phrase boundary marks that are present in the participant response. In implementing the 4 second window, it was discovered that the machine learner perhaps overfit

for the participants that marked phrases very closely together. On the other side of this issue, a 4 second window was not large enough for the participants that marked phrases very far apart. So, the size of the window should be determined by the phrase durations. In creating the target array from Participant 3's data, it was discovered that there were more 1's than 0's. This contributed to a less-accurate prediction for the machine learner, contrary to the accuracy and log loss metrics computed, since these metrics were measured in comparison to Participant 3's target array. Computing the specificity metric exploited this issue, as well.

Data Splitting

For the logistic regression model, only training and testing sets were created. The training set consisted of 85% of the total data, and the testing data consisted of 15% of the total data. These proportions were found to produce the highest accuracy, when compared to splitting the data 80-20 or 90-10 (train-test). A validation set was not considered for the logistic regression model since it had no effect on training the model, and because there was only a small amount of data available. For the sequential neural network, a validation set was created to assist in training the model. For this model, the training set consisted of 80% of the total data, the validation set consisted of 10% of the total data, and the testing set consisted of 10% of the total data. The accuracy on the validation set was checked every epoch to ensure that the model was not overfitting. For the model to stop training, the validation accuracy would have had to decrease over the previous 10 epochs. Splitting the data 80-10-10 led to better accuracy than 70-15-15 or 75-10-15 (train-validation-test).

Model Size and Optimization

The sequential neural network consisted of two hidden layers, both containing 3 units and both using the hyperbolic tangent (\tanh) activation function. There were other activation functions considered, such as ReLU (rectified linear unit), ELU (exponential linear unit), and Sigmoid, just to name a few. Using more and less than 3 units per layer was also explored, but 3 facilitated a smoother training process. The number of hidden layers was also explored; one, two and three were used, but two yielded the best results. For larger models with more data for training and testing, more hidden layers with more different activation functions and more units should be considered.

The optimization algorithm used for the sequential neural network is the Adam (Adaptive Moment Estimation) optimizer. The Adam optimizer incorporates elements from the Adadelta algorithm and RMSprop method by computing adaptive learning rates for each parameter (Ruder, 2016). The Stochastic Gradient Descent algorithm was also considered for the model with different learning rates and momentum, but the Adam optimizer was ultimately chosen because it allowed for earlier convergence.

Section 4.2: Future Work

The work done in this study leaves the door open for continued study. It is advised that more acoustic features be considered when training a model for this task. It seemed as though the acoustic features utilized in this study were useful for training small models, but one next step would be to build more sophisticated models. This would require more data in the form of human responses to stimuli.

4.2.1 Improvements to the Machine Learning Models

There are many ways in which the methods and results of this study can be improved.

Outlined in this section are a few improvements to the machine learning models that would aid in future amendments to this study's work.

More Acoustic Features

More acoustic features could inform the machine learner to consider different patterns in the signal that were otherwise ignored because of the use of so few acoustic features. For the three acoustic features used in this study (intensity, spectral flatness, and rhythmic density), the task was simply to look for troughs since, from the Phrase Detection Algorithm, we know that the participant responses generally correlated to a trough in the measure of each acoustic feature. For machine learners, it is helpful to encompass many features that represent a different pattern for input vectors (Prabhat, 2017). Had there been more acoustic features in the input vector, the machine learner likely could have found many different patterns in each feature that aided in the perception of phrase boundaries. It is advised that future endeavors of this work incorporate many more acoustic features to capture the variety of objective aspects of the sound.

More stimuli

The models ultimately need more data to perform better. One way to obtain more data is to add more pieces to the experiment. The experiment could have involved less participants, and had each of the participants listen to more pieces. Since the input vector given to the machine

learner consists of as many data points as there are fourths of seconds in the stimuli, more pieces would have created more data without having to recruit more participants for the experiment.

The variety of stimuli could also be expanded to incorporate more styles of composition. In this study, according to the participant responses, pieces written after 1850 had higher variance in phrase durations. This could mean that phrasing in music of the Western Classical tradition written after 1850 is more difficult to discern. Pieces written before 1850 tend to use more conventional harmonies and tonalities. Phrase boundaries are thus much easier to detect for those earlier works. Using a balance of early and late works should improve the predictability of the model for many styles of music.

Different Models

This study only utilized two models--a logistic regression model and a feedforward neural network--although, there are many other types of models that could do quite well for this problem. One approach would be a Long Short Term Memory network (LSTM). An LSTM is a special type of recurrent neural network that can handle long-term dependencies. In attempting to predict phrase boundaries, it makes intuitive sense that the model should remember what has happened in the past, so it can make an informed prediction. It is my intention for the foreseeable future to implement an LSTM to perform this task.

The models used and described here are categorized as supervised learning, since the classification labels are known to the machine learner and the researcher. A different approach to this task could be reinforcement learning, which dispenses rewards based on a model's predictions. The relationship between the action and the reward is unknown to the model. The

idea of considering a reinforcement learning approach is driven by (Roederer, 2008) as described in Section 1.2. Since the brain's limbic system works by dispensing either reward or punishment, it is conceivable that the same mechanisms that cause a human to discern phrases in music could inform a machine learning agent to make predictions based on the feedback from an artificial limbic system.

4.2.2 Possible Application

Consider a music teacher demonstrating a musical idea to a student. This music teacher might want to give a visual depiction of a certain way of phrasing that a student is not quite grasping, and the aid of another mental representation can provide a more definite understanding of the musical language. This project is concerned with the language of music. Phrasing is supposedly very subjective (Olsen, 2016), but even through the subjectivity of music in phrase structures, there are certain aural patterns that humans find to be more pleasing than others. One could argue that not all humans agree upon what phrase structures sound pleasing, and that may be the case, but there is something to be said about a world-class musician's interpretation versus an amateur's interpretation. The application of these models could use the phrasing of the world-class musician to demonstrate a better understanding of the musical language to the aspiring amateur musician.

Section 4.3: Conclusion

In this project, we defined phrasing in music as the way in which a musical passage is expressed to relay auditory information. We showed that the segmentation of phrases into phrase

boundaries becomes critical for understanding how humans hear and interpret this information (Olsen, 2016; Roederer, 2008). Human desire to understand phrases; the motivation for understanding phrases, even when there is no apparent circumstantial need, is linked to a neurological reward in the limbic system of the brain (Roederer, 2008). The desire to understand phrases extends to developing computational techniques for quantifying this auditory information.

Bibliography

- [1] Olsen, K. N., Dean, R. T., & Leung, Y. (2016). What constitutes a phrase in sound-based music? A mixed-methods investigation of perception and acoustics. *PLoS one*, 11(12), e0167643.
- [2] Knösche, T. R., Neuhaus, C., Haueisen, J., Alter, K., Maess, B., Witte, O. W., & Friederici, A. D. (2005). Perception of phrase structure in music. *Human Brain Mapping*, 24(4), 259-273.
- [3] Gingras, B., Pearce, M. T., Goodchild, M., Dean, R. T., Wiggins, G., & McAdams, S. (2016). Linking melodic expectation to expressive performance timing and perceived musical tension. *Journal of Experimental Psychology: Human Perception and Performance*, 42(4), 594.

- [4] Glushko, A., Steinhauer, K., DePriest, J., & Koelsch, S. (2016). Neurophysiological correlates of musical and prosodic phrasing: shared processing mechanisms and effects of musical expertise. *PLoS one*, 11(5), e0155300.
- [5] Roederer, J. G. (2008). *The physics and psychophysics of music: An introduction*. Springer Science & Business Media.
- [6] Degara, N., Pena, A., Davies, M. E., & Plumley, M. D. (2010, March). Note onset detection using rhythmic structure. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 5526-5529). IEEE.
- [7] Prabhat, A., & Khullar, V. (2017, January). Sentiment classification on big data using Naïve Bayes and logistic regression. In *2017 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.
- [8] Peeters, G., Giordano, B. L., Susini, P., Misdariis, N., & McAdams, S. (2011). The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5), 2902-2916.
- [9] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

- [10] Koelsch, S., Gunter, T. C., Wittfoth, M., & Sammler, D. (2005). Interaction between syntax processing in language and in music: An ERP study. *Journal of cognitive neuroscience*, 17(10), 1565-1577.
- [11] Silva, S., Branco, P., Barbosa, F., Marques-Teixeira, J., Petersson, K. M., & Castro, S. L. (2014). Musical phrase boundaries, wrap-up and the closure positive shift. *Brain research*, 1585, 99-107.
- [12] Hung, Y. N., Chen, Y. A., & Yang, Y. H. (2018). Learning disentangled representations for timber and pitch in music audio. arXiv preprint arXiv:1811.03271.
- [13] Boersma, Paul & Weenink, David (2020). Praat: doing phonetics by computer [Computer program]. Version 6.1.14, retrieved 2 May 2020 from <http://www.praat.org/>
- [14] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, Oriol Nieto, Josh Moore, Dan Ellis, et al. Librosa: v0.4.0. Zenodo, 2015. <https://doi.org/10.5281/zenodo.18369>
- [15] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 12, 2825-2830 (2011)

[16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[17] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272.

[18] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke & Travis E. Oliphant. Array programming with NumPy, Nature, 585, 357–362 (2020), DOI:10.1038/s41586-020-2649-2

[19] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55

[20] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[21] Peirce, J. W., Gray, J. R., Simpson, S., MacAskill, M. R., Höchenberger, R., Sogo, H., Kastman, E., Lindeløv, J. (2019). PsychoPy2: experiments in behavior made easy. Behavior Research Methods. 10.3758/s13428-018-01193-y

[22] Naxos. “Daily Download: Gregorian Chant for Good Friday - Hymnus.” Classical MPR, 10 Apr. 2020,
www.classicalmpr.org/story/2020/04/10/daily-download-gregorian-chant-for-good-friday--hymnus.

[23] Musopen. “Mozart Symphony no. 40 in G minor, K. 550.”

<https://musopen.org/music/1577-symphony-no-40-in-g-minor-k-550/>

[24] Naxos. “Daily Download: Ludwig Van Beethoven - Piano Sonata No. 21, Op. 53

‘Waldstein’: I. Allegro Con Brio: Your Classical.” YourClassical, 4 Nov. 2015,

www.yourclassical.org/story/2015/11/04/daily-download-ludwig-van-beethoven--piano-sonata-no-21-op-53-waldstein-i-allegro-con-bri.

[25] Naxos. “Daily Download: Anton Bruckner - Symphony No. 7: I. Allegro Moderato: Your

Classical.” YourClassical, 28 June 2019,

www.yourclassical.org/story/2019/06/28/daily-download-anton-bruckner--symphony-no-7-i-allegro-moderato.

[26] Internet Archive. “Frederic Chopin - Ballade No. 1 in G Minor, Op. 23 (From The Pianist) :

Free Download, Borrow, and Streaming.” Internet Archive, 20 Dec. 2013, 14:54:17,

archive.org/details/FredericChopinBalladeNo.1InGMinorOp.23FromThePianist.

[27] Naxos. “Daily Download: Sergei Rachmaninoff - Vocalise in E Minor: Your Classical.”

YourClassical, 5 May 2017,

www.yourclassical.org/story/2017/05/05/daily-download-sergei-rachmaninoff--vocalise-i-n-e-minor.

Appendices

Appendix 1: Experiment

Appendix 1A: IRB Approval Letter

Appendix 1B: Verbal Instructions for Experiment

Appendix 1C: Consent Form

Appendix 1D: Participant Questionnaire

Appendix 1E: Recruitment Email

Appendix 1F: CITI Program Human Subject Research Training Certificate

Appendix 1G: Debriefing Statement

Appendix 2: Plots of Acoustic Features

Appendix 2A: Logarithm of Intensity Plots

Appendix 2B: Acoustic Onsets Plots

Appendix 2C: Spectrograms

Appendix 2D: Marked Intensity Plots

Appendix 2E: Marked Spectral Flatness Plots

Appendix 2F: Marked Rhythmic Density Plots

Appendix 1: Experiment

Appendix 1A: IRB Approval Letter

Bard College

Institutional Review Board

Date: October 20, 2020

To: Evan Petratos

Cc: Sven Anderson, Deborah Treadway, Brandt Burgess

From: Tom Hutcheon, IRB Chair

Re: A Machine Learning Approach to the Perception of Phrase Boundaries in Music

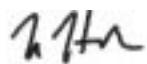
DECISION: APPROVED

Dear Evan,

The Bard Institutional Review Board has reviewed your revisions and approved your proposal entitled “A Machine Learning Approach to the Perception of Phrase Boundaries in Music.” Your proposal is approved through October 20, 2021 and your case number is 2020OCT20-PET.

Please notify the IRB if your methodology changes or unexpected events arise.

This sounds like a really interesting project and we wish you the best of luck with your research!



Tom Hutcheon
IRB Chair
thutcheo@bard.edu

Appendix 1B: Verbal Instructions for Experiment

Verbal description of Consent Process and Experimental Procedure

Brief Overview and Procedure

Thank you for agreeing to participate in this study. Your participation is expected to take about 45 minutes. During this time, you'll intently listen to each stimulus two times through, consecutively. The first listen is more for you to become familiar with the style and the perceived phrase lengths. You will be asked to focus on the ends of phrases that occur throughout the piece. During this experiment, you will be interacting with a PsychoPy python program which plays each stimulus and acts as a stopwatch when pressing the spacebar. You should use the program to mark the ends of phrases in both listens for each stimulus. You can adjust your volume as needed, for comfort.

Safety and Confidentiality

If you're ever uncomfortable for any reason and would like to stop participating, that is OK, just say so. You will still be paid for your time. The computer that is running the python program will be sanitized before you use it. I will be physically distanced from you as you listen to the stimuli. Your data will be recorded and imported into an Excel spreadsheet, and will be marked by an anonymous identifier (e.g. Participant 1, Participant 2, etc.). Your data and your responses will remain confidential (i.e. there will be no association with your name to your data).

Consent

Before we start you need to read this consent form carefully. Consent forms are necessary so that you are accurately informed about the research process and you understand your rights. If you have any questions, just ask me. Then, if you agree with the content of the consent form, sign at the bottom. If you'd like a copy of the consent form, you can take a copy with you.

Appendix 1C: Consent Form

Consent to participate in this experiment

Project Title: A Machine Learning Approach to the Perception of Phrase Boundaries in Music

Researcher: Evan Petratos

Faculty Adviser: Sven Anderson

I am a student at Bard College and I am conducting experiments for my Senior Project. I am studying the phrase boundaries in music from an objective and computational perspective.

During this study, I will ask you some questions as to your musical background and hearing abilities, as well as your past experiences with computation, if any. This experiment is designed to last approximately 45 minutes. Experiments will take place in the Reem and Kayden Center (RKC) at Bard College.

Potential risks of participation include annoyance or boredom with the music being presented. If I present you with a piece that you do not want to listen to or feel uncomfortable listening to, please tell me and we will move on to the next piece or stop the experiment. All necessary safety precautions against COVID-19 are implemented during the experiment (i.e. masks wore and physical distancing). In addition, the computer, keyboard, desks and headphones will be sanitized between participants, and I (the researcher) will maintain participant contact information for 14 days (in order to aid in contact tracing measures).

Participants will be paid for their time. While no direct benefits to participants are expected, participants may receive indirect benefits from learning about the research process, as well as about the background motivating the present work.

All the information you provide will be confidential. I will use pseudonyms when I write about this research. I will keep my experiment notes and data secure in a password-protected file on my personal computer and in an external hard drive kept in a secure location. Only my faculty adviser and I will have access to this information.

Participant's Agreement

I understand the purpose of this research. My participation in this interview is voluntary. If I wish to stop the experiment for any reason, I may do so without having to give an

explanation.

The researcher has reviewed the relevant risks and potential direct/indirect benefits with me, to the extent there are any. I am aware the information will be used in a Senior Project that will be publicly accessible online and at the Stevenson Library of Bard College in Annandale, New York. I have the right to review, comment on and withdraw information prior to November 30, 2020.

The information gathered in this study is confidential with respect to my personal identity. I understand that complete confidentiality cannot be guaranteed, since the researcher may be required to surrender notes and/or data if served with a court order.

If I have questions about this study, I can contact the researcher at ep9407@bard.edu or the faculty adviser at [sanderson@bard.edu](mailto:sanderso@bard.edu). If I have questions about my rights as a research participant, I can contact the chair of Bard's Institutional Review Board at irb@bard.edu.

I have been offered a copy of this consent form to keep for myself.

I am at least 18 years of age and I consent to participate in today's experiment.

Participant's signature

Date

Participant's printed name

Researcher's signature

Appendix 1D: Participant Questionnaire

Participant's Questionnaire

Most questions require yes/no answers, but elaboration is encouraged.

Section 1: Hearing Impairments

Do you have to strain when you are conversing?

Do you have to strain when you are listening to music?

Do you have difficulty when someone speaks in a whisper?

Do you have difficulty discerning nuances in quiet music?

Do you often find that loud noises cause discomfort?

Section 2: Musical Background

Have you taken private music lessons before?

Do you have at least three years of formal musical experience?

Do you come from a background that encourages musical excellence?

What genres of music do you consider to be your favorites?

Do you listen to classical music, as rooted in the traditions of Western culture, regularly?

Section 3: Computation and Physics

Have you taken any classes in computer science or physics? If so, please specify.

Are you familiar with concepts of machine learning?

Are you familiar with basic concepts of acoustics as a branch of physics?

Appendix 1E: Recruitment Email

Participant Scouting Email

To be sent to: Conservatory students at Bard College

Subject:

Call for Participants in Senior Project Experiment

Body:

Hi all,

I am in need of musicians to participate in an experiment. This experiment is intended to gather human responses to distinguish phrases in music. The experiment is not a psychological study to understand how humans hear music. The data will be used to train and test a computer algorithm to become familiar with humans' interpretation of phrasing.

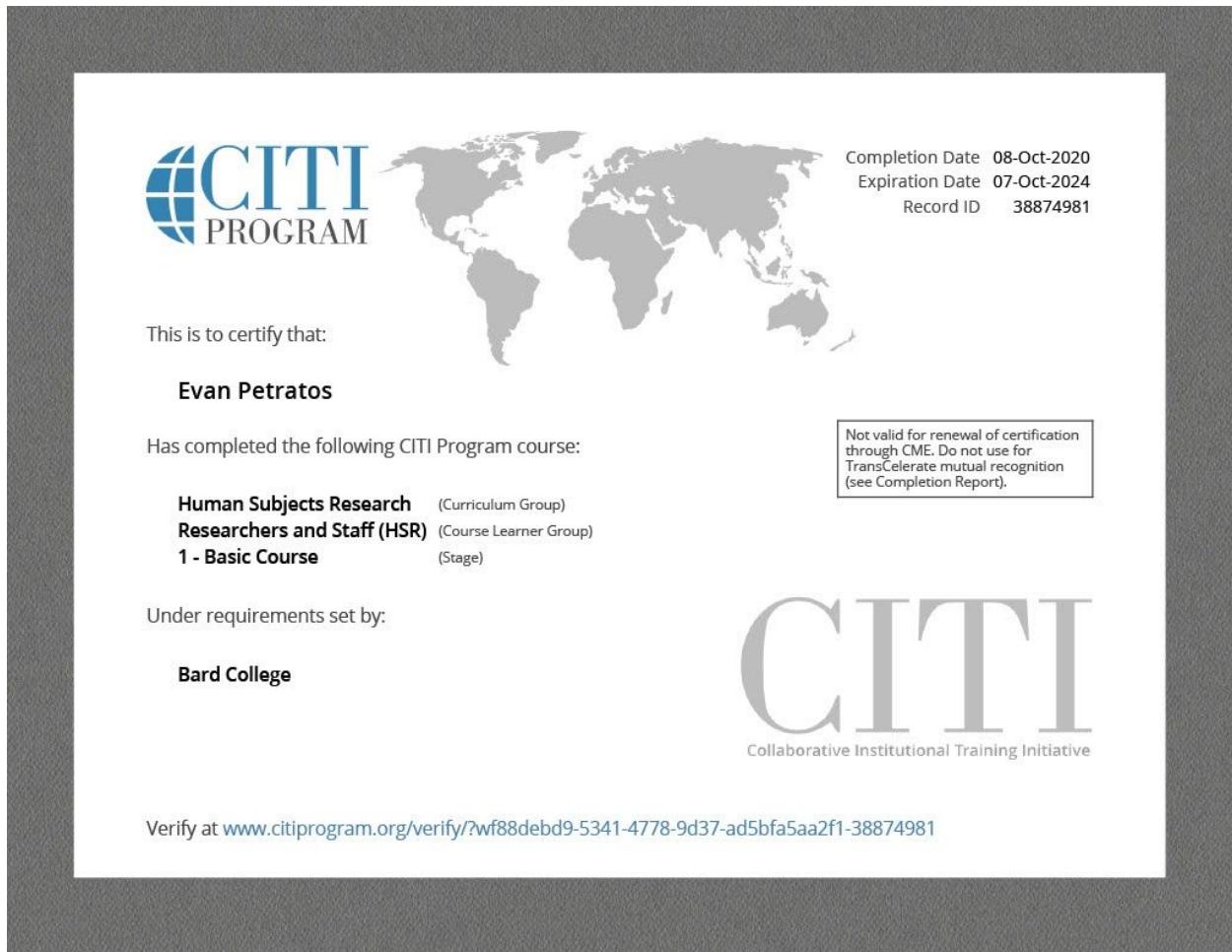
As a participant in this experiment, you will be asked to intently listen to excerpts from pieces of the Western Classical tradition. While listening, you will be marking the perceived phrase endings in every instance you feel appropriate. The experiment is expected to take about 45 minutes. You will be paid for your time.

If you would like to participate, or if you would like more information about the study, please send me an email (ep9407@bard.edu).

Thank you,

Evan Petratos

Appendix 1F: CITI Program Human Subject Research Training Certificate



Appendix 1G: Debriefing Statement

Debriefing Statement

Brief Introduction: The intention of this project is to create a working, efficient algorithm that can detect phrase boundaries in music from a given audio file. The acoustic features involved in detecting phrases from a given audio file include spectral analysis of timbre, waveform analysis of amplitude or loudness, and an analysis of frequency. In each case, the goal is to look for common patterns that occur between “events” during an audio file [Olsen]. For data collection and correctness of the algorithm, such an event will be determined by a willing participant who intently listens to certain pieces of music and marks where phrases end. The set of data collected will serve a training and testing data for the computer algorithm.

Phrasing: Phrasing in music is thought to be subjective [Olsen]. The artistic decisions that drive a musician to create a phrase a certain way, with a certain length, is always at the discretion of that musician, led by conventional harmonic and melodic progressions. Phrasing organizes auditory information in speech and music [Olsen]. From a physics point of view, speech and music both follow an uneven acoustic flow, and this flow is partitioned into structures that we identify as phrases [Knösche]. It is natural for humans to perceive the inflection in music or language [Olsen, Knösche, Glushko].

Human Language/Link to Speech: The musical phrase is a means of expression, similar to language. Human language has a natural rise and fall in contour, which conveys meaningful information. Language is formed with words, clauses, sentences, etc. which can have an inflection, pauses, articulation, and mood. All of these attributes of language are used to enunciate and articulate the speaker’s message. Many, or perhaps all, of these attributes are integral in the expression of a musical message. Music and speech are related in this way, and the previous endeavors on the analysis of human language should be considered in the analysis of musical expression. Both speech and music carry acoustic information which is intended to be interpreted by the human brain. In the case of human language, it is perhaps obvious that the sound produced carries information. Music can perhaps be considered the co-product of the evolution of human language. In the evolution of hominids, operations of sound processing, analysis, storage, and retrieval became necessary for the development of human speech [Roederer].

Phrase Boundaries: Phrase boundaries have proven to be discernible and identifiable in speech, but there is much less study and literature concerning phrasing in music. For humans discerning boundaries in speech segments (particularly words, clauses, and sentences), there are noticeable acoustic cues to allow for this discernment. These acoustic cues occur as changes in intensity/amplitude, rhythmic pattern, pitch contour

and many other objective acoustic variables. Phrase boundaries become a central component for the perception of the structures that are created from an uneven acoustic flow in speech and in music [Knösche]. The acoustic variables that concern the segmentation of human speech are the same variables that discern phrase boundaries in music [Knösche]. Because we can structure auditory information in this way, the boundaries of phrases are of utmost importance for the analysis of how humans interpret information in language and in music [Olsen, Roederer]

Acoustic Background: Advancements in human speech progressed the reception of music and the perception of subjective sensations of timbre, consonance, tonal expression, sense of resolution, and the long-term structures of melodic lines. The perception of these sensations is linked to limbic rewards in the search for phonetic content of sound that can be identified as logical manifestations of acoustical signals. The limbic system works in sort of a “binary” way; it dispenses either reward or penalty, which are emotional states of the brain. The motivation to listen to, analyze, and store musical sounds, even when there is no apparent circumstantial need, triggers a feeling of pleasure—this limbic reward. To facilitate information processing in speech, the motivation emerged to understand acoustical signals and receive emotional feedback [Roederer].

References for further reading

- Olsen, K. N., Dean, R. T., & Leung, Y. (2016). What constitutes a phrase in sound-based music? A mixed-methods investigation of perception and acoustics. *PloS one*, 11(12), e0167643.
- Knösche, T. R., Neuhaus, C., Haueisen, J., Alter, K., Maess, B., Witte, O. W., & Friederici, A. D. (2005). Perception of phrase structure in music. *Human Brain Mapping*, 24(4), 259-273.
- Gingras, B., Pearce, M. T., Goodchild, M., Dean, R. T., Wiggins, G., & McAdams, S. (2016). Linking melodic expectation to expressive performance timing and perceived musical tension. *Journal of Experimental Psychology: Human Perception and Performance*, 42(4), 594.
- Glushko, A., Steinhauer, K., DePriest, J., & Koelsch, S. (2016). Neurophysiological correlates of musical and prosodic phrasing: shared processing mechanisms and effects of musical expertise. *PloS one*, 11(5), e0155300.
- Roederer, J. G. (2008). *The physics and psychophysics of music: An introduction*. Springer Science & Business Media.

Appendix 2: Plots of Acoustic Features

Appendix 2A: Logarithm of Intensity Plots

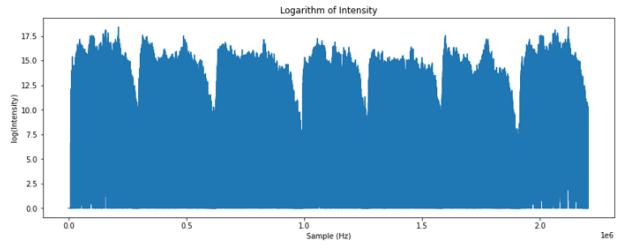


Figure 2A1: Logarithm of Intensity of the Gregorian Chant excerpt.

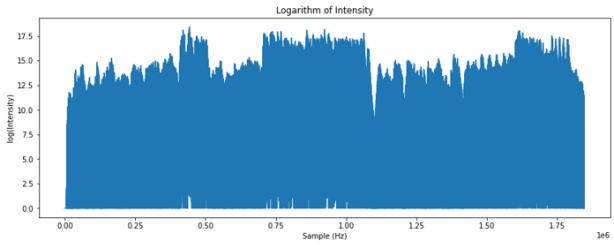


Figure 2A2: Logarithm of Intensity of the first excerpt from Mozart's Symphony No. 40.

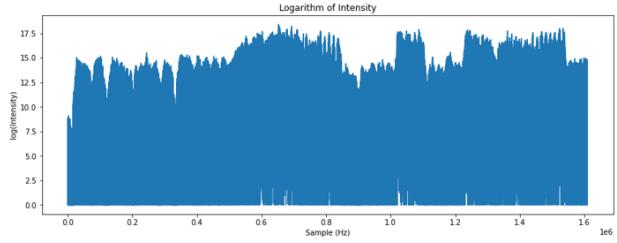


Figure 2A3: Logarithm of Intensity of the second excerpt from Mozart's Symphony No. 40.

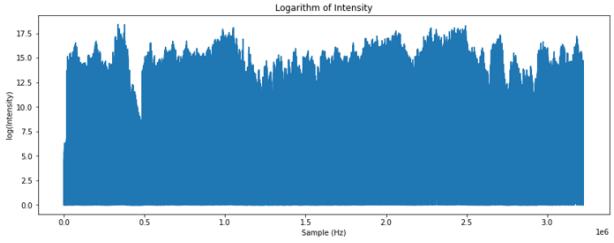


Figure 2A4: Logarithm of Intensity of the excerpt from Beethoven's Waldstein Piano Sonata.

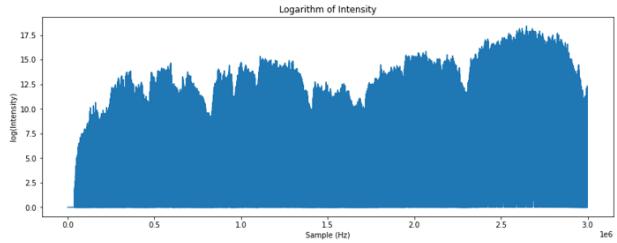


Figure 2A5: Logarithm of Intensity of the first excerpt from Bruckner's Symphony No. 7.

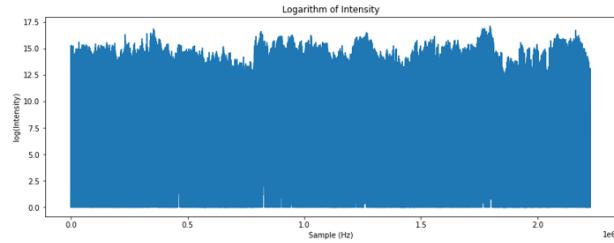


Figure 2A6: Logarithm of Intensity of the second excerpt from Bruckner's Symphony No. 7.

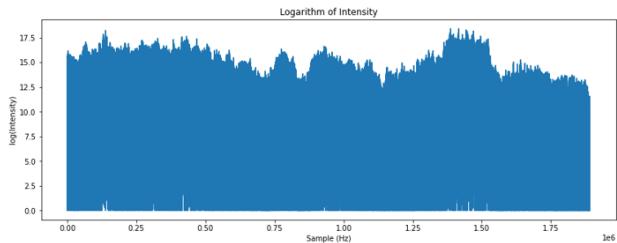


Figure 2A7: Logarithm of Intensity of the third excerpt from Bruckner's Symphony No. 7.

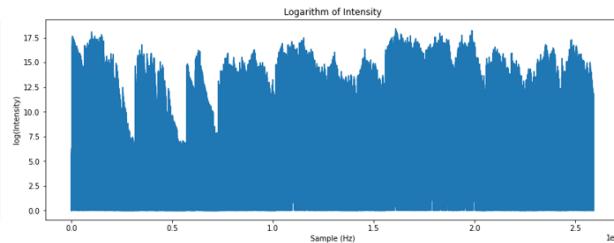


Figure 2A8: Logarithm of Intensity of the first excerpt from Chopin's Ballade No. 1.

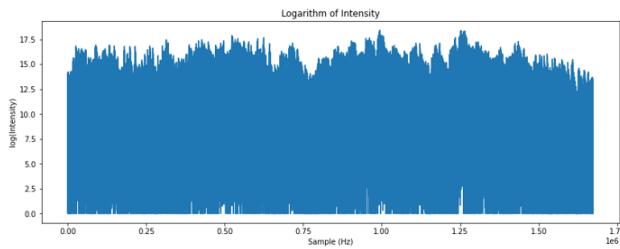


Figure 2A9: Logarithm of Intensity of the second excerpt from Chopin's Ballade No. 1.

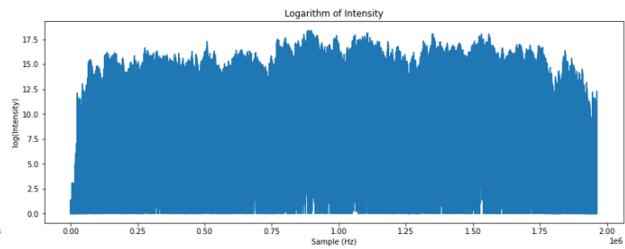


Figure 2A10: Logarithm of Intensity of the excerpt from Rachmaninoff's Vocalise.

Appendix 2B: Acoustic Onsets Plots

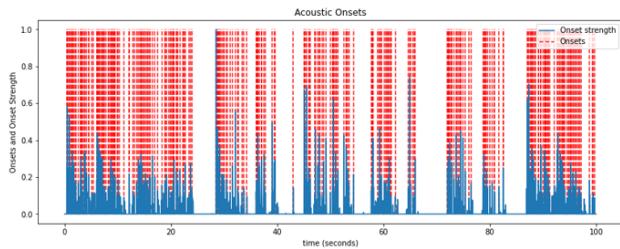


Figure 2B1: Plot of Acoustic Onsets of the Gregorian Chant excerpt.

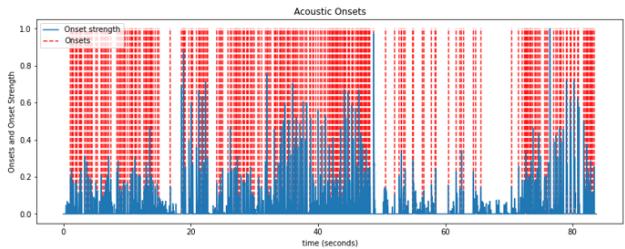


Figure 2B2: Plot of Acoustic Onsets of the first excerpt from Mozart's Symphony No. 40.

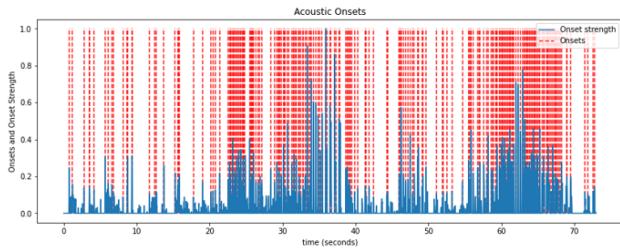


Figure 2B3: Plot of Acoustic Onsets of the second excerpt from Mozart's Symphony No. 40.

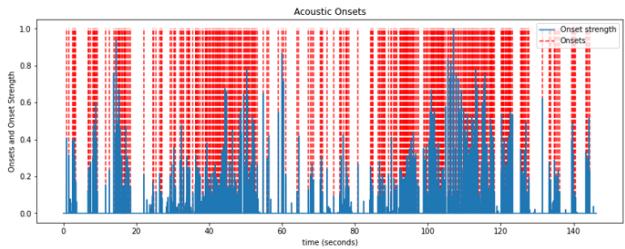


Figure 2B4: Plot of Acoustic Onsets of the excerpt from Beethoven's Waldstein Piano Sonata.

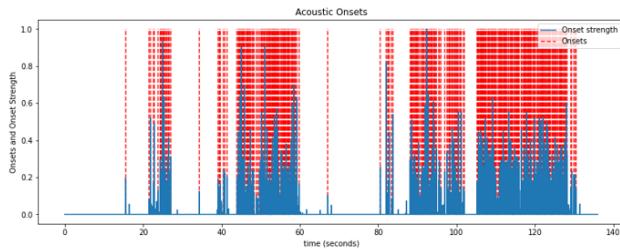


Figure 2B5: Plot of Acoustic Onsets of the first excerpt from Bruckner's Symphony No. 7.

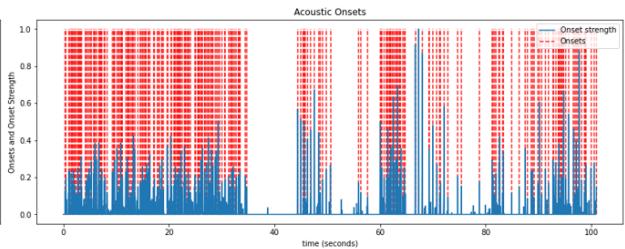


Figure 2B6: Plot of Acoustic Onsets of the second excerpt from Bruckner's Symphony No. 7.

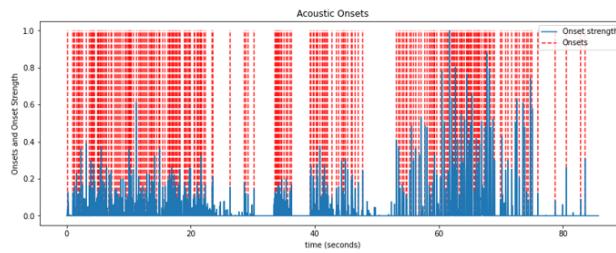


Figure 2B7: Plot of Acoustic Onsets of the third excerpt from Bruckner's Symphony No. 7.

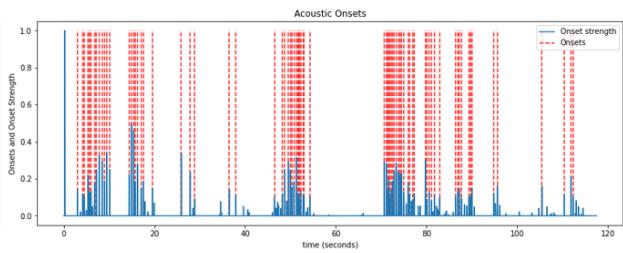


Figure 2B8: Plot of Acoustic Onsets of the first excerpt from Chopin's Ballade No. 1.

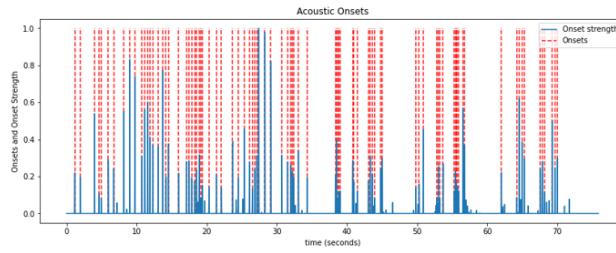


Figure 2B9: Plot of Acoustic Onsets of the second excerpt from Chopin's Ballade No. 1.

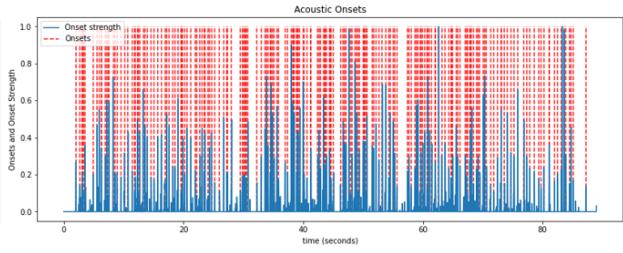


Figure 2B10: Plot of Acoustic Onsets of the excerpt from Rachmaninoff's Vocalise.

Appendix 2C: Spectrograms

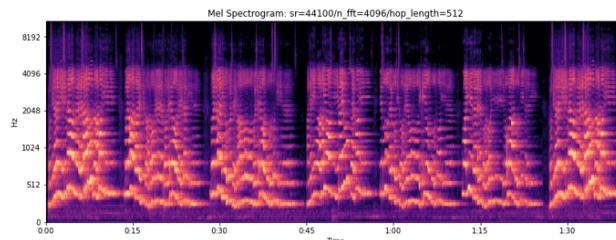


Figure 2C1: Spectrogram of the Gregorian Chant excerpt.

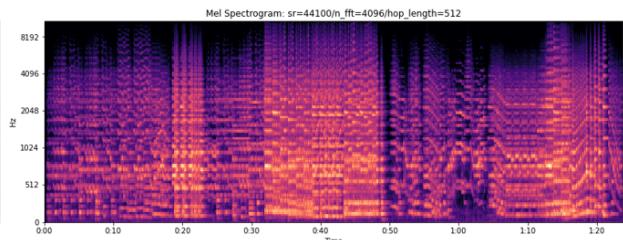


Figure 2C2: Spectrogram of the first excerpt from Mozart's Symphony No. 40.

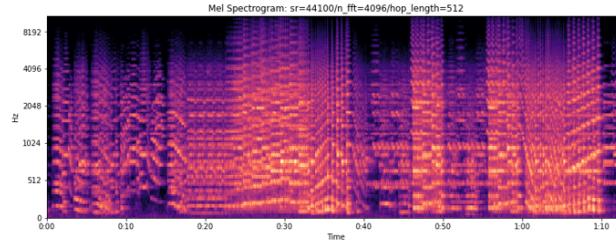


Figure 2C3: Spectrogram of the second excerpt from Mozart's Symphony No. 40.

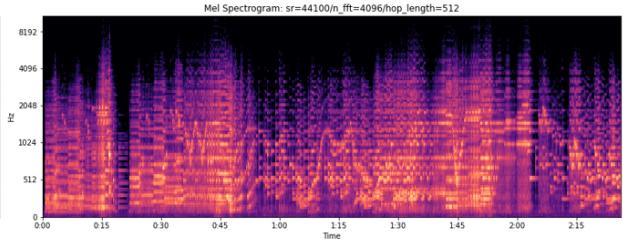


Figure 2C4: Spectrogram of the excerpt from Beethoven's Waldstein Piano Sonata.

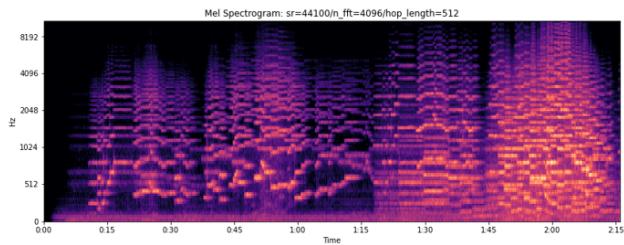


Figure 2C5: Spectrogram of the first excerpt from Bruckner's Symphony No. 7.

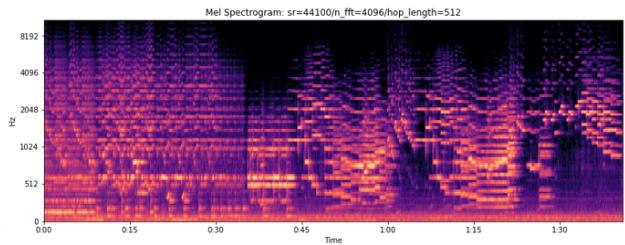


Figure 2C6: Spectrogram of the second excerpt from Bruckner's Symphony No. 7.

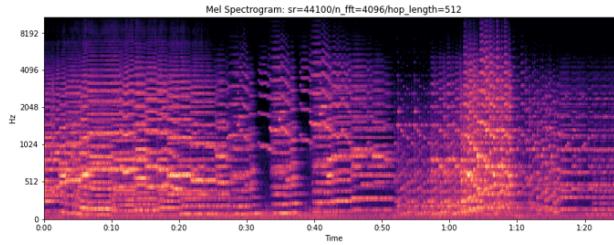


Figure 2C7: Spectrogram of the third excerpt from Bruckner's Symphony No. 7.

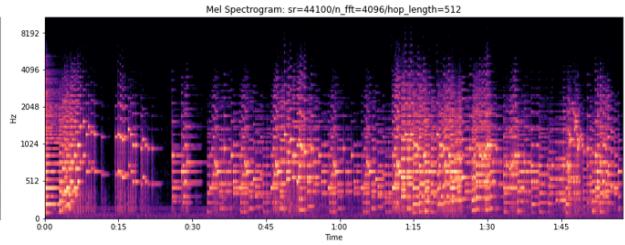


Figure 2C8: Spectrogram of the first excerpt from Chopin's Ballade No. 1.

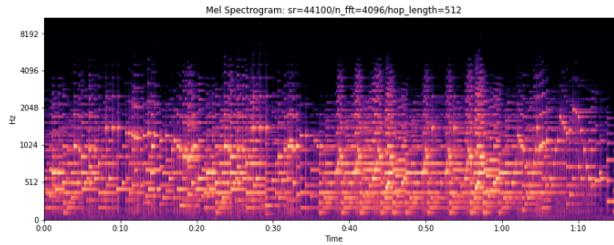


Figure 2C9: Spectrogram of the second excerpt from Chopin's Ballade No. 1.

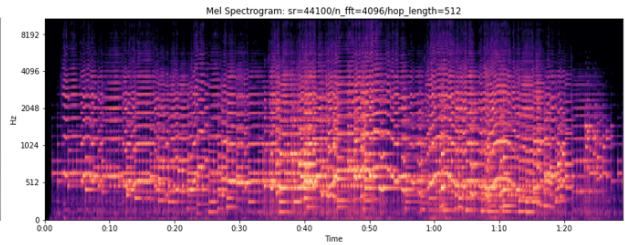


Figure 2C10: Spectrogram of the excerpt from Rachmaninoff's Vocalise.

Appendix 2D: Marked Intensity Plots

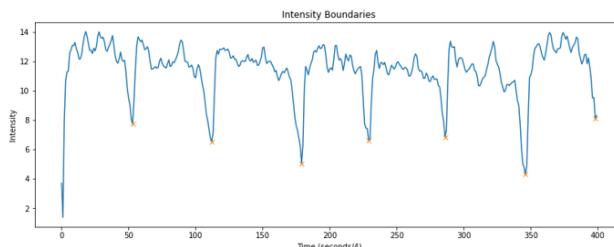


Figure 2D1: Marked plot of the intensity of the Gregorian Chant excerpt.

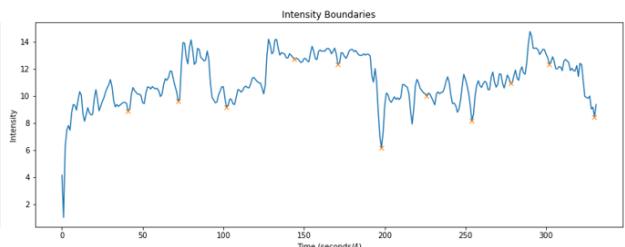


Figure 2D2: Marked plot of the intensity of the first excerpt from Mozart's Symphony No. 40.

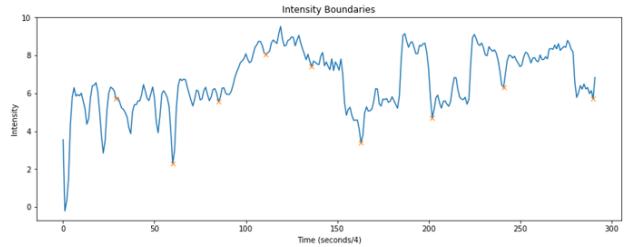


Figure 2D3: Marked plot of the intensity of the second excerpt from Mozart's Symphony No. 40.

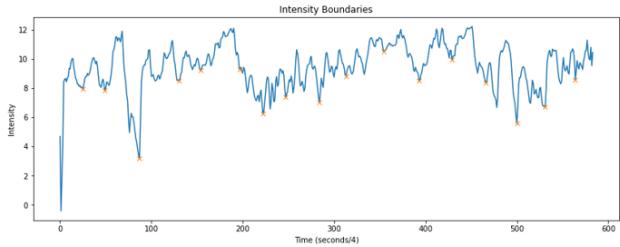


Figure 2D4: Marked plot of the intensity of the excerpt from Beethoven's Waldstein Piano Sonata.

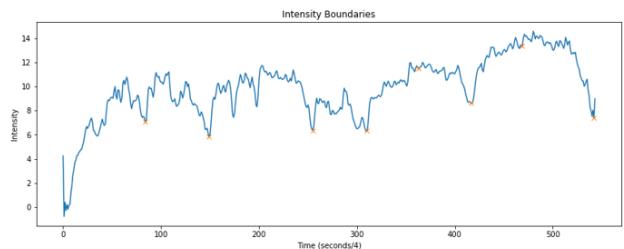


Figure 2D5: Marked plot of the intensity of the first excerpt from Bruckner's Symphony No. 7.

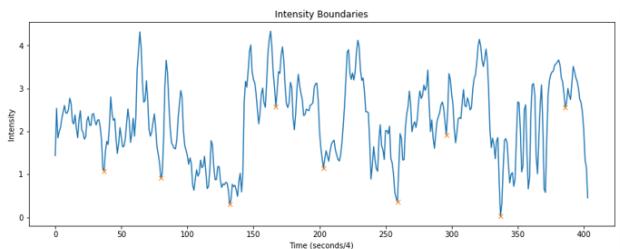


Figure 2D6: Marked plot of the intensity of the second excerpt from Bruckner's Symphony No. 7.

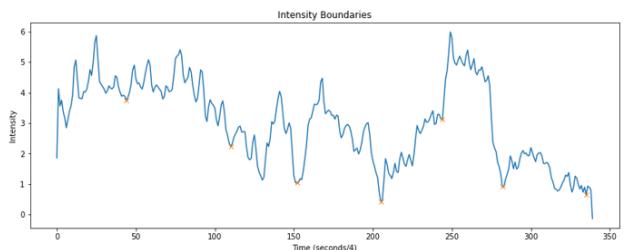


Figure 2D7: Marked plot of the intensity of the third excerpt from Bruckner's Symphony No. 7.

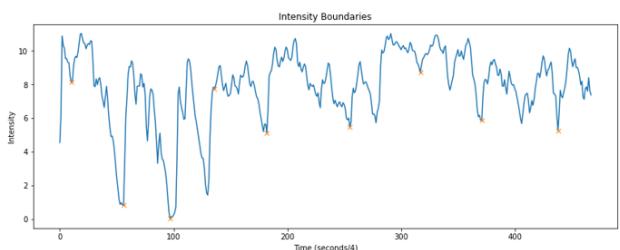


Figure 2D8: Marked plot of the intensity of the first excerpt from Chopin's Ballade No. 1.

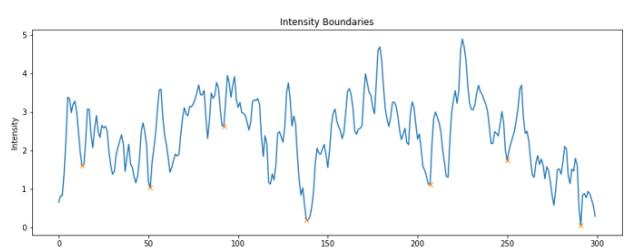


Figure 2D9: Marked plot of the intensity of the second excerpt from Chopin's Ballade No. 1.

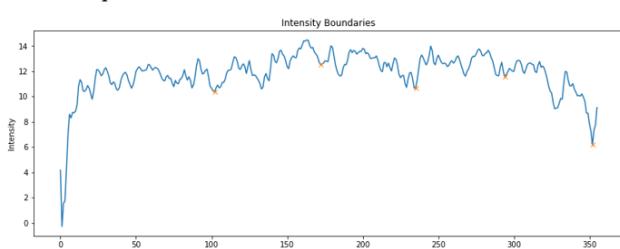


Figure 2D10: Marked plot of the intensity of the excerpt from Rachmaninoff's Vocalise.

Appendix 2E: Marked Spectral Flatness Plots

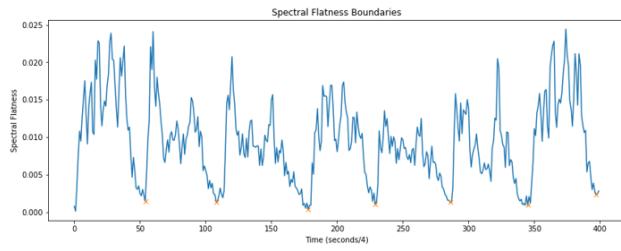


Figure 2E1: Marked plot of the spectral flatness of the Gregorian Chant excerpt.

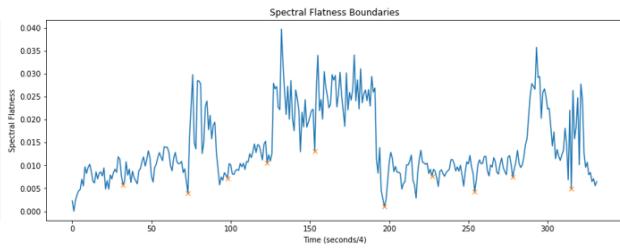


Figure 2E2: Marked plot of the spectral flatness of the first excerpt from Mozart's Symphony No. 40.

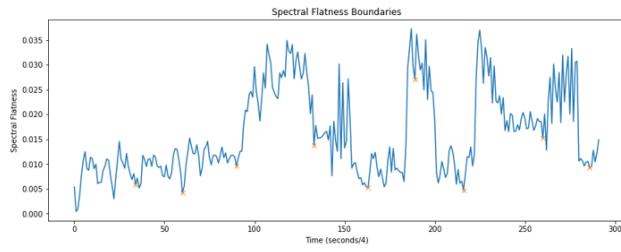


Figure 2E3: Marked plot of the spectral flatness of the second excerpt from Mozart's Symphony No. 40.

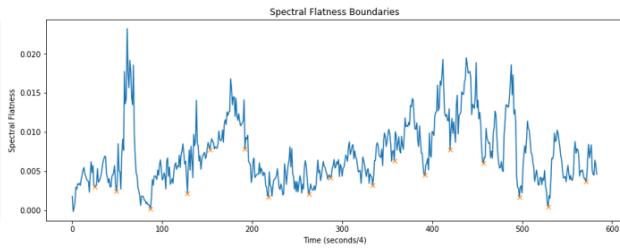


Figure 2E4: Marked plot of the spectral flatness of the excerpt from Beethoven's Waldstein Piano Sonata.

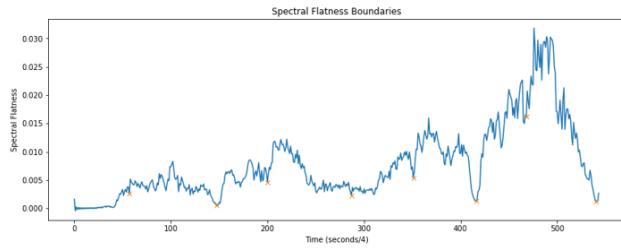


Figure 2E5: Marked plot of the spectral flatness of the first excerpt from Bruckner's Symphony No. 7.

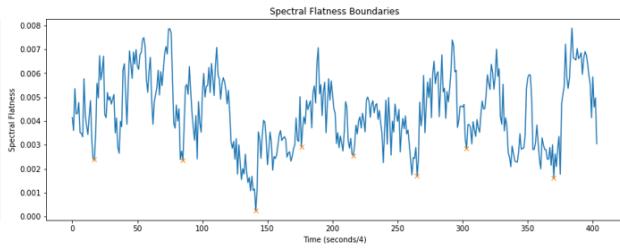


Figure 2E6: Marked plot of the spectral flatness of the second excerpt from Bruckner's Symphony No. 7.

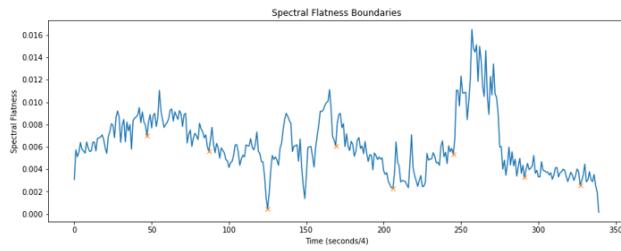


Figure 2E7: Marked plot of the spectral flatness of the third excerpt from Bruckner's Symphony No. 7.

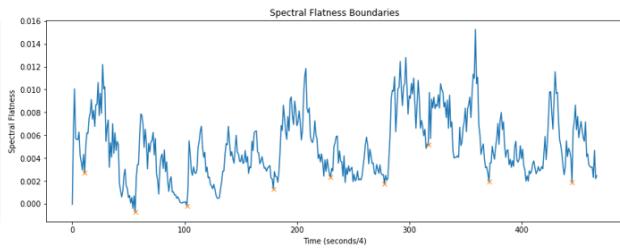


Figure 2E8: Marked plot of the spectral flatness of the first excerpt from Chopin's Ballade No. 1.

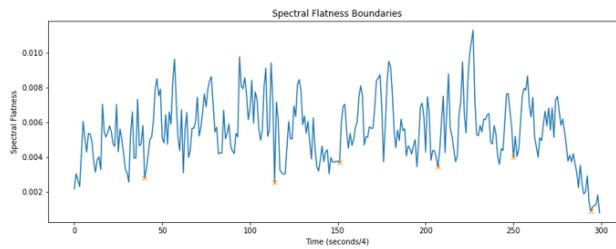


Figure 2E9: Marked plot of the spectral flatness of the second excerpt from Chopin's Ballade No. 1.

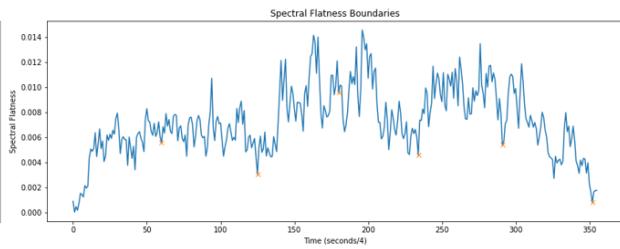


Figure 2E10: Marked plot of the spectral flatness of the excerpt from Rachmaninoff's Vocalise.

Appendix 2F: Marked Rhythmic Density Plots

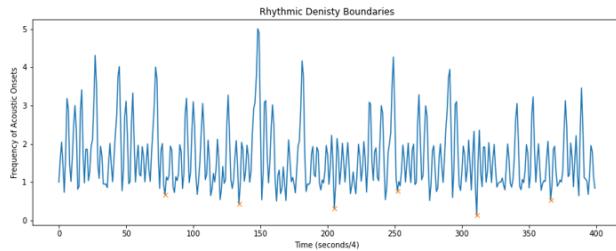


Figure 2F1: Marked plot of the rhythmic density of the Gregorian Chant excerpt.

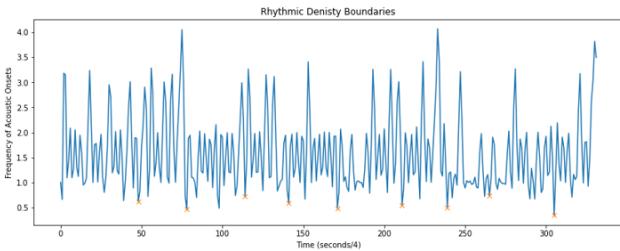


Figure 2F2: Marked plot of the rhythmic density of the first excerpt from Mozart's Symphony No. 40.

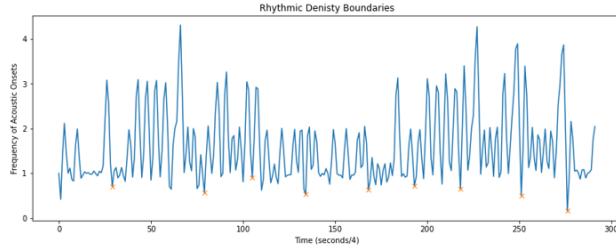


Figure 2F3: Marked plot of the rhythmic density of the second excerpt from Mozart's Symphony No. 40.

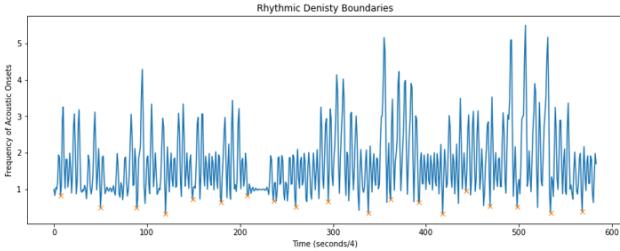


Figure 2F4: Marked plot of the rhythmic density of the excerpt from Beethoven's Waldstein Piano Sonata.

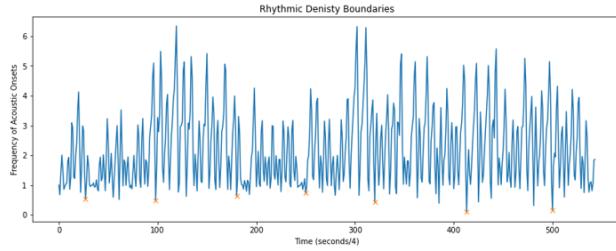


Figure 2F5: Marked plot of the rhythmic density of the first excerpt from Bruckner's Symphony No. 7.

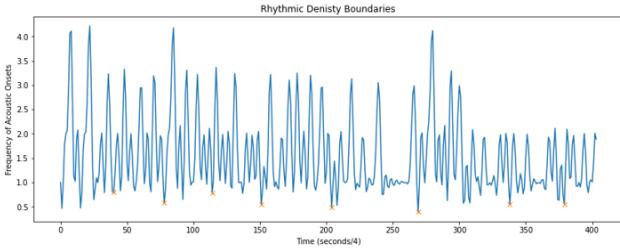


Figure 2F6: Marked plot of the rhythmic density of the second excerpt from Bruckner's Symphony No. 7.

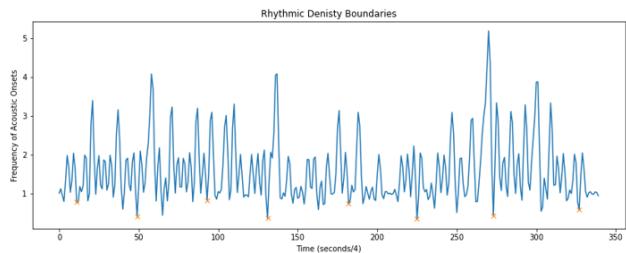


Figure 2F7: Marked plot of the rhythmic density of the third excerpt from Bruckner's Symphony No. 7.

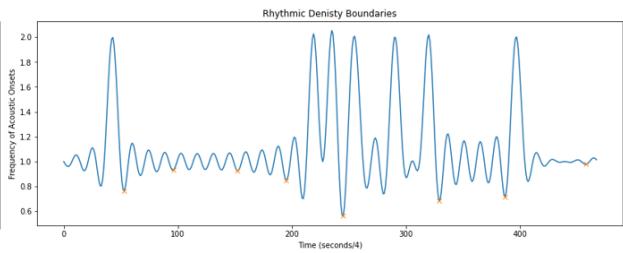


Figure 2F8: Marked plot of the rhythmic density of the first excerpt from Chopin's Ballade No. 1.

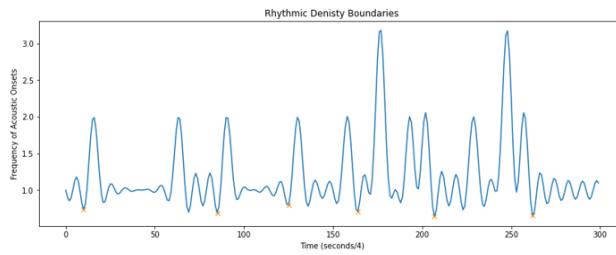


Figure 2F9: Marked plot of the rhythmic density of the second excerpt from Chopin's Ballade No. 1.

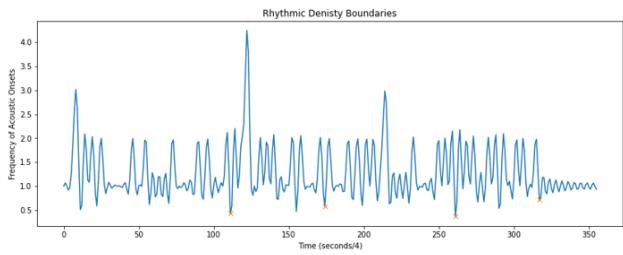


Figure 2F10: Marked plot of the rhythmic density of the excerpt from Rachmaninoff's Vocalise.