Music as an Information-Bearing Medium: Extracting Latent Structure for Theoretical and Practical Purposes

by

Ladan Mahabadi

School of Computer Science

McGill University, Montreal

August 2013

A thesis submitted to McGill University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

To my parents,

To Lida,

To Mom Badri, and
To an ephemeral tornado.

Acknowledgement

I would like to start by thanking my supervisors, Professors Daniel J. Levitin and Doina Precup for their support and guidance over the past two years. I am forever grateful for your support, direction, and for allowing me to explore and learn about anything and everything that time permitted.

I would also like to thank Professor Claude Crépeau for his undying belief in me; without his faith in me, I would not be where I am today. Throughout my journey at McGill, I have learned from many individuals , whom I have encountered during my various *random walks*. Though not all encounters have been perfect, I have learned from each and every one. My McGill years mark a great chapter in my life, and it would not be as colorful without my friends, the great professors at McGill (Thank you Patrick, Luc, Adrian, Prakash and Hamed), the fantastic library system (Thank you for allowing this book worm to read about anything and everything fathomable), and other brilliant individuals whose paths have crossed mine. I'd like extend my gratitude to Mr. Brian McMillan at the Schulich Music library for his great assistance .

I'd like to thank Associate Provost Dean Martin Kreiswirth and Professor Xiao-Wen Chang for their insightful support at my time of crisis. I consider myself fortunate to be

a part of such a diverse, creative group of intellectuals and I have benefited tremendously, both personally and academically from their insights, challenges and great company. The administration staff at the School of Computer Science have also played an instrumental role in this chapter of my life. I am, in particular, grateful to Diti Anastasopoulos, Sheryl Morrissey, Andrew Bogecho , Kailesh Mussai, and Ron Simpson — among many more that I may have neglected to mention. I'd like to thank Kota Nakamura and Sten Thomson for their invaluable assistance in gathering biographical information on Western composers used in my analyses. Finally, I'd like to thank Sara Florence and Jordana Saks.

This thesis has been a journey of self discovery; I have loved every minute of learning at McGill. I have learned the true merits of tenacity, meticulous curiosity, *random samplings*, and *mind tangos*. To my friends, thank you for all the memories, great walks, intellectually invigorating rants leading to epiphanies, late emails, good sips of coffee cherished while snobbishly frowning on shabby coffee, mind-clarifying hikes, skydives, and flying lessons. You have made my life in Montreal richer. This thesis has been enriched by feedback and help from Amin Ranjbar, Shahe Shnorhokian, Karle-Philip Zamor (thank you for everything, including the translation), Grayden Wagner, Adrian Kaats, Gjergji Hasa, Kota Nakamura, and Kamal Zellag.

I am truly lucky to have an incredible family: Thank you mom, dad, and Lida for believing in *cBskt* (despite all her stunts); thank you for supporting me intellectually and otherwise through it all. A special "thank you" is reserved for our stubborn Fergie, for always making sure that I was "up 'n running" by 5 am.

As I have often said, words are rarely sufficient, so Let $X =$ "Thank you", then "Multiply X by ∞, take it to the depth of forever, and you will still have barely a glimpse of [how grateful I am for it all.]"(William Parrish)

Abstract

Music is an information-bearing medium, containing highly structured signals. Its complexity derives from its information-rich structure and enables music to be molded to the imagination of any creator, regardless of time or geography. This thesis investigates the rich information encoded in the temporal structure of music. It represents rhythm as a progression of musical event durations (a note or a silence), and uses similarities hidden in different layers of musical rhythm to construct a structural identity (temporal scale-free features). These temporal scale-free structures are used for longitudinal analysis of compositions in a composer's oeuvre, and are applied as composer descriptors for classification in two large Western and non-Western music score repertoires. That is, this work presents the scale-free structural features in rhythm as a universal, information-rich feature set.

More precisely, temporal scale-free correlation exponents are computed for two major music categories: Western and non-Western compositions. Collections containing these categories — 1165 and 1498 music pieces respectively — are further sub-categorized by composer name (Western Collection) and geographical region or country of origin (non-Western Collection). The analysis provides a more granular, composer-by-composer validation of the scale-free structure of rhythm in the Western collection. This indicates that such exponents vary for composers of different musical eras, distinct time periods and various countries of origin. Moreover, for a particular composer, though there are key similarities between different compositions' temporal scale-free exponents, they exhibit non-negligible variation during the course of the composer's professional life. These variations can account for the high accuracy in the classification results for (1) Western composers from disparate eras or with distinct compositional styles (e.g., Gershwin versus Mozart); (2) non-Western regions of distinct musical style and heritage (e.g., Chinese versus Persian music); and (3) Western versus non-Western pieces.

In the second part of this thesis, rich information in music and particular features of human musical cognition are used to expand the existing CAPTCHA paradigm — wherein problems that are easy for humans but which cannot yet be solved efficiently by computer programs are used to distinguish between humans and automated music analysis programs (computational Turing test). The ubiquity of music across all cultures, its complex structure at different layers, and the perceptual characteristics and limitations of the human auditory system are leveraged to construct more accessible, aesthetically pleasing and secure computational Turing tests. Music-based CAPTCHAs, called mCaptchas, are introduced to improve Web accessibility for individuals with visual impairments, to help avoid susceptibility to security flaws of existing audio CAPTCHAs, and to improve the overall user experience. The mCaptcha scheme generates musical streams which can be heard as two by humans, but which appear as a single inseparable unit of music to the state-of-the-art automated programs. A user is presented with a contextual question based on a popular song which the user can identify in the composite music stream (challenge). The security of the scheme lies in the fact that state-of-the-art music analysis programs cannot yet solve the challenge correctly before the mCaptcha expires.

Empirical evidence of the scheme's security is presented for over 2000 mCaptchas, while its usability is tested by approximately 500 individuals on the Amazon Mechanical Turk (AMT) online market. These results demonstrate that humans can efficiently and accurately solve the generated music-based challenges while sophisticated computer programs fail.

Abrégé

La musique est un support d'information contenant des signaux hautement structurés. La complexité de la musique découle de la richesse informative de sa structure qui lui permet de se conformer à l'imaginaire de chaque créateur, quelle que soit l'époque ou la localisation géographique. La présente thèse examine la richesse informative dissimulée dans la structure temporelle de la musique. Cette thèse envisage le rythme comme une progression de la durée d'événements musicaux (une note ou un silence) et utilise les similarités dissimulées dans les différentes couches de rythme musical pour construire une identité structurelle (caractéristiques temporelles libres d'échelle). Ces structures temporelles libres d'échelle sont utilisées pour une analyse longitudinale d'un échantillon des compositions faisant partie de l'œuvre d'un compositeur. Elles sont également appliquées comme des descripteurs de l'œuvre d'un compositeur en vue d'une classification des compositions en deux grands répertoires, occidental et non-occidental. C'est-à-dire, ce travail présente les caractéristiques structurelles libres d'échelle du rythme comme un ensemble de caractéristiques universelles riches en information. Plus précisément, ces exposants de la corrélation temporelle libre d'échelle sont calculés dans deux catégories: compositions occidentales et non-occidentales. Ces catégories, contenant respectivement 1165 et

1498 pièces musicales, sont ensuite divisées en sous-catégories, par nom du compositeur (collection occidental) et sa région ou son pays d'origine (collection non-occidentales). Cette analyse fournisse une validation plus granulaire, compositeur par compositeur, de la structure rythmique libre d'échelle dans le répertoire occidental. Cela indique que de tels exposants varient pour les compositeurs de différentes époques musicales, différentes périodes et différents pays d'origine. De plus, malgré l'existence de similarités majeures entre les exposants temporels libres d'échelle des compositions d'un compositeur, ces exposants varient d'une manière significative au cours de la vie professionnelle. Ces variations peuvent expliquer la grande précision des résultats de la classification pour (1) les compositeurs occidentaux venant de différentes époques ou présentant des styles de composition différents (par ex., Gershwin versus Mozart); (2) les régions non-occidentales caractérisées par un style et un héritage musical particulier (par ex., musique chinoise versus musique perse); et (3) les morceaux de musique occidentale versus non-occidentale.

Dans la deuxième partie de la présente thèse, la richesse informative de la musique et les caractéristiques particulières de la cognition musicale humaine sont utilisées afin d'élargir le paradigme CAPTCHA existant. Dans ce paradigme, des problèmes facilement résolus par les humains, mais que les programmes informatiques ne parviennent pas encore à résoudre, sont utilisés pour distinguer la manière dont ces problèmes sont résolus par les humains et les programmes d'analyse de la musique automatisés (test de Turing). L'ubiquité de la musique à travers toutes les cultures, la structure très complexe des différentes couches qui la constituent, ainsi que nos connaissances des caractéristiques et des limitations de la perception auditive humaine permettent aujourd'hui de construire des tests de Turing plus accessibles, plus agréables esthétiquement et mieux sécurisés. Je

présente les mCAPTCHAs, CAPTCHAs basés sur la musique pour répondre à plusieurs objectifs: améliorer l'accessibilité au Web pour les individus souffrant de handicaps visuels, diminuer la vulnérabilité des CAPTCHAs audio existants aux brèches de sécurité, et améliorer l'expérience générale de l'utilisateur. Le schéma mCAPTCHA génère des flux musicaux pouvant être décomposés en 2 pièces musicale distinctes par ségrégation auditive chez les humains, tout en étant considérés comme des unités musicales monolithiques par les programmes automatisés actuels. Un utilisateur reçoit une question contextuelle basée sur une chanson populaire incluse dans ce flux musical généré (défi). La sécurité du schéma réside dans le fait que les programmes actuels les plus sophistiqués ne sont pas encore capables de résoudre correctement le défi avant l'expiration du mCAPTCHA.

L'évidence empirique de la sécurité de ces schémas est présentée pour plus de 2000 mCAPTCHAs, tandis que sa convivialité a été testée par environ 500 individus sur le marché en ligne Amazon MechanicalTurk. Ces résultats démontrent que les humains peuvent efficacement et correctement résoudre les défis musicaux générés, là où les logiciels les plus sophistiqués échouent.

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

"There is nothing more difficult than talking about music."

Charles-Camille Saint-Saëns

Music is a form of communication, a critical part of human cultural evolution [Mil00, Cro01], and a fundamental medium of communication and emotional expression [BT99]. It transcends borders, languages and eras [BBN95, MDA$^+$03].

The scientific study of musical communication includes research on music cognition, the creation of music, and its emotional and therapeutic effects. Despite the use of massive computational power and music analysis tools, there is still much to be understood about how humans create and comprehend music. The composer Igor Stravinsky noted, "music's exclusive function is to structure the flow of time and keep order in it" [SS86]. Music follows a progression of sounds and silences, which form audible patterns, variations and themes over time [Mey56, Ber76, Ric00]. The particular repetitions — either exact or with variation [Ric00] — form a structural identity that is rich with information. When music is heard as a contiguous audio stream, the listener builds expectations for what is to be heard next [Ric00, KNH$^+$05], and the consequent surprise or validation is key to the overall emotional musical experience [Mey56, Til08]. In other words, structural repetitions are a significant factor in the listeners' experience of and appreciation for smusic [Sch00].

In the first part of this thesis, I investigate a concise temporal representation of structural repetitions in Western and non-Western music, and apply the extracted structural information to classification. The concise representation consists of power law temporal features generated from the rhythmic structure of music. Features are computed using the long-term correlations (fluctuation analysis in the time-domain) and power spectra (spectral analysis) from a simple temporal representation of music. If the probability that a particular feature has value k is proportional to $k^{-\alpha}$, for sufficiently large k, then it has a power law exponent of α. In our context, the power law correlation exponents are defined as α , where the probability distribution of the correlations decays proportionally to $L^{-\alpha}$, where L is a parameter such as length of the correlation blocks. To investigate information latent in the structural repetitions of Western and non-Western music, I analyze the power law temporal features generated from the rhythmic structure of 1165 movements from compositions of Western classical music and 1498 pieces of non-Western music. This Western analysis includes music by 24 composers from the $16^{\mathtt{th}}$ to the $20^{\mathtt{th}}$ century, and the non-Western corpus consists of music pieces from Africa, China, Iran and Turkey. Finally, the computed temporal scale-free exponents are used as descriptors in binary classifications of various Western composers, as well as .

The second part of this thesis focuses on a music-based computational access control scheme. Alan Turing [Tur50] introduced a simple test that would differentiate between machines and humans. He designed it to further quantify what it means for a machine to be said to have intelligence indistinguishable from that of a human and it has come be known as the Turing test. Luis von Ahn [vABHL03] extended the notion of a Turing test from a philosophical one to a security tool used to stop automated attacks and spurious registrations online. This invention — called a CAPTCHA[1] — was introduced as an access control security measure. CAPTCHAs, ubiquitous on the web, use problems that are believed to be much easier for humans to solve compared to computers. The original design was textual and obfuscated a short text with noise and other visual distortions. Two main characteristics of a CAPTCHA are security and usability. Security in this context is defined as the computational difficulty of a machine breaking the system by passing the test, and usability measures how easy it is for a human to correctly solve the test. Various forms of the scheme have been devised including audio CAPTCHAs. Existing audio CAPTCHAs consist of a set of scrambled (English) letters and numbers embedded in a noisy background. The user has to identify the letters and type them correctly (the challenge) [vABHL03]. However, this challenge is not sufficiently difficult for the current state-of-the-art machine learning and signal processing algorithms. Hence, audio CAPTCHAs were shown to be insecure [BB09, BBP$^+$11]. Moreover, from a human usability point of view, both text-based and audio CAPTCHAs are still lacking in providing an effective and pleasant Turing test [BBF$^+$10].

I introduce a novel music-based Completely Automated Public Turing test to Tell Computers and Humans Apart, denoted by *mCaptchas*, by applying music to generate computationally secure tests that distinguish between humans and machines. The new approach is an efficient and computationally secure challenge-response scheme. Here, security, contrary to the existing paradigm, does not rely on obfuscation but rather on using certain features particular to human musical cognition.

[1] CAPTCHA stands for: “Completely Automated Public Turing test to tell Computers and Humans Apart,” and is a trademark of Carnegie Mellon University.

1.1 Overarching Themes

This thesis analyzes structured information in music, and considers two related computational cases. In the first, structured information is used to classify music; the second case focuses on cases where machine algorithms cannot classify music. Although both parts investigate structured information latent in music, the first half uses music as represented in scores (symbolic music), and the second half uses music as it is performed. The research presented in this dissertation started with the fractal analyses of rhythm in music. mCaptchas were the result of a thought experiment pondering the existence of a music-based question, which is computationally intractable or impractical. While designing the mCaptcha system, I realized that the computational indistinguishability of the two music streams interleaved can be amplified if the two streams were chosen to have structural similarities. In other words, knowing that the machines will have to perform some form of analysis on the structure of the mCaptcha stream, the security of my proposed system would be improved if the overall structure does not reveal two distinct signatures. That is, if the mCaptcha system could select its music streams such that they have similar temporal scale-free signatures, then the computational cost of distinguishing between two such interleaved streams would even be higher. In this way, the analysis of scale-free temporal structures (1/f) latent in music grew organically out of my efforts to make the mCaptacha design as rigorous as possible, and in turn, the results of the 1/f work can be applied to the mCaptcha work.

1.2 Thesis Contributions

This thesis lies at the intersection of two fields of research: music cognition and machine learning. It draws from research in both fields to extract latent structural information in music for both theoretical and practical purposes. The first component of this thesis focuses on the temporal structure of music and studies temporal self-similarities in various compositions by extracting correlation power law exponents. This work comprises two novel contributions. It shows how such power law exponents in music can be used to classify compositions by their composer, musical era, and geographic origin. It also investigages the efficacy of using such exponents to classify the genre of music as Western vs. non-Western. The results show that the emergence of scale-free patterns in music is not limited to particular geographical, cultural or historical time periods.

The second part of the thesis focuses on another way of using latent information in musical structure, in this case to create user-friendly security interfaces in the form of mCaptchas, a new music-based computational Turing test. The integration of music into this framework is novel. The analysis includes security challenges and a large-scale human usability tests that use crowdsourcing. The experiments indicate that not only are the mCaptchas secure, but also effective for human use. mCaptchas improve Web accessibility for individuals with visual impairments , and provide a more pleasnt human experience due to the integration of music and not noise.

1.3 Organization

The body of this thesis is organized as follows. Chapter 2 introduces definitions and methods used to compute the self-similarity power law exsponents for musical compositions. Chapter 3 studies self-similarity exponents for a large corpus of Western compositions, and Chapter 4 performs a similar analysis for a corpus of Chinese, Persian, African and Turkish music. In Chapter 5, I present mCaptchas by first motivating the need for this new design. I proceed by highlighting its features , outlining the implementation , and demonstrating its advantages over existing CAPTCHA designs. The thesis concludes with a summary of the contributions in Chapter 6. An outline of possible future extensions of this research is also discussed in the same chapter.

2

Background

"Music is given to us with the sole purpose of establishing an order in things, including, and particularly, the coordination between man and time."

- Igor Stravinsky [Str36]

This chapter introduces basic notions needed for temporal analysis of music. It also outlines the methodology used to analyze the power law features in rhythm . The Western music collection is organized into groups each labeled by the name of the composer, and the non-Western collection is categorized by country or region of origin. I refer to these labels as *composer* labels. The atomic unit of analysis is a movement of a particular composition from a certain composer, which is uniquely identified by its name and the name of its composer. In the case of non-Western compositions, compositions are identified by their origin.

2.1 Composers and their Compositions

Bregman defines an audio stream as "the perceptual unit that represents a single happening ... [which] serves the purpose of clustering related qualities." [Bre94] Accordingly, I denote a temporal sequence of inter-onset durations for a movement by a *musical stream*, and represent it as a time series — a sequence of values ordered in time. A musical stream is a sequence of events that display continuity and consistency [MB79, Bre94]. In the context of my analyses, the series is consistent as it originates from one movement — a self-contained portion of a composition with a unified interpretation or theme [def] — and its continuity originates from its temporal order, which creates its identity [MB79]. Let $p_j^{c_i}$ represent the $j^{\texttt{th}}$ musical stream composed by *composer* c_i. In this thesis, c_i is a label associated with a group of musical streams that have either been composed by the same individual (e.g., Beethoven), or are associated with a particular geographical region (e.g., China). I denote a consistent collection of these composer labels, $c_1, \ldots, c_n$, by $C = \{c_i\}_{i=1}^{n}$. To analyze the temporal regularities that are characteristic of each composition, the occurrences of musical events (notes or silences) are used to compute inter-onset durations.

A musical stream consists of combinations of notes and silences ordered in time in a meaningful way, and *patterns* are similarities found between particular elementary components of the stream. Order matters: music *identity* is determined by the temporal structure — particular order of notes and silences — and changes made to this structure result in a different identity [Ler01, Pat03].

A music time series is calculated by computing the time intervals in between adjacent musical events. The durations are measured in standardized quarter note lengths. That is, a quarter note will be encoded as 1, a whole note as 4, and an eighth note as 0.5. These durations, in my analyses, are extracted from symbolic music using `music21` — an open-source toolkit for computer-aided musicology [CA10]. Symbolic music is a score-based representation of music that encodes information about notes, various voices and other musical symbols.

2.2 Redundancy, Predictability and Variation in Music

> "*Indeed, all figures of speech, and all metaphors, in speech and music alike, depend ultimately on repetition, which is then subjected to variation, or as the linguists say, transformation*"
>
> - Leonard Bernstein [Ber76]

There exist multiple parallels between music and language [Ber76, LM03, Pat03]. Both are means of communication and use temporal patterns to convey coherency and meaning [Kru91, Kru00]. Words are the building blocks of languages, and sequences of words, ordered in time, form phrases and other components of languages [Akm01]. Languages are filled with repetitions of such building blocks.

Patterns in music — repeated individual or groups of notes or silences in a particular order [Ler01, Pat03, LM03] — are redundancies appearing over time in a musical stream. Levitin and Menon define musical structure as "that which exists when musical elements are bound, through temporal coherence, in a way that leads to informational redundancy and expectation." [LM03] In the following sections, I further elaborate on key notions of redundancy and variation, as applicable to this thesis.

2.2.1 Measures of Predictability

A key concept in information theory is the notion of Shannon entropy, which quantifies the amount of information gained after an observation is made. Shannon defines redundancy as a measure of the statistical constraints imposed by the language on the text being analyzed [Sha51]. He highlights the statistical constraints of a language — that for instance, requires a `C` to be followed by an `E` with a higher probability than a `C` followed by a `Z` in English . Such constraints directly influence the patterns emerging in the derived words. He also defines entropy to be the amount information, on average, learned from each letter of a source , which outputs words constructed from a particular finite alphabet. Put differently, entropy measures how many bits, on average, are needed for encoding a particular word, and can be directly associated with average information contained therein. Redundancy and randomness are two sides of the same coin. High redundancy entails low information — as almost everything is predictable.

More precisely, consider a model that generates the musical durations — measured in standardized quarter note lengths which constitute the music time series — one at a time. That is, assume that there is a probabilistic generator that generates the raster, represented as, $\langle X_1 \ldots X_L \rangle$, where each X_i is independent and identically distributed. The entropy of the source, $H(X)$, quantifies the predictability of the X_is: if the raster is highly predictable, then learning X_i will not add much to what is already known from$\langle X_1 \ldots X_{i-1} \rangle$. In other words, when entropy is low, predictability will be high, which means that predicting future duration events, based on partial information, is more accurate. Autocorrelation, or similarity between a time series and itself at various time lags, is a measure of redundancy. The following basic definitions are used in my analysis, and I include them here for the sake of completion. For more rigorous treatment of these notions, the interested reader is referred to the wealth of literature available on statistical tools in time series analysis.

A time series $X(n) = \langle x_1, \ldots, x_n \rangle = [x_i]_{i=1}^n$ is an ordered series of observations. For instance, the number of record sales for a particular band in the course of twenty years can be represnted by $X(20) = \{x_1, \ldots, x_{20}\} = \{x_i\}_{i=1}^{20}$, where x_i denotes the record sales in year i. An observed time series $X(n)$ can be considered a realization of a generative random process $X = [X_i]_{i=1}^n$ with an associated probability function, P. The *sample mean* of a time series, $X(n)$ is denoted by: $\mu(X(n)) = \frac{1}{n}\Sigma_{i=1}^n x_i$. I use the sample mean as an estimate for the expected value of the generating random process, $E[X] = \Sigma_{x \in L} x \Pr\{X_i = x\}$ [GS01]. Spread and variation are computed using the standard deviation:

Definition 2.2.1 (Moments [GS01])**.** *The* $k^{\texttt{th}}$ *moment of* X *is defined as* $m_k = E[X^k]$*, and the* $k^{\texttt{th}}$ *central moment* σ_k *is defined similarly after removing the expected value:* $\sigma_k = E[(X - \mu)^k]$.

In Definition 2.2.1, k is typically a positive integer; however, when it is a fraction, it denotes fractional moments of order k. The mean and variance, two key characteristics of a time series, can be reformulated in terms of moments as well:

- (*M*ean) $k = 1 : m_1 = \mu = E[X]$,
- (*V*ariance) $k = 2 : \texttt{var}(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$,
- (*S*tandard deviation) $\sigma = \sqrt{\texttt{var}(X)}$.

A time series $X(n)$ is *strictly stationary* if all of its moments - including $E[X(n)]$, $\texttt{var}[X(n)]$, and higher moments - are time-independent. More precisely, the joint statistical distribution of $X_{t_1}, \dots, X_{t_l}$ is the same as the joint statistical distribution at lag τ, $X_{t_{1+\tau}}, \dots, X_{t_{l+\tau}}$ [Pri, Nas06]. A more practical notion is that of a *weak stationarity* which implies that the mean and the variance do not depend on time, and that self-similarity — measured through autocorrelation — will not depend on time but only on time lag [Pri, Nas06].

Definition 2.2.2 (Sample Autocorrelation)**.** *Let* $X(n) = \{x_1, \dots, x_m\}$ *denote a time series. The autocorrelation function at lag* $l \in \{0, \dots, m-1\}$ *is defined as:*

$$\mathtt{AC}(l) = \frac{\Sigma_{i=l+1}^{m}(x_i-\mu(X(n)))(x_{i-l}-\mu(X(n)))}{\Sigma_{i=1}^{m}(x_i-\mu(X(n)))^2}$$

The sample autocorrelation measures how similar a time series is to itself. The more repetitive a time series is, the higher the product of time-lagged portions of the time series, which leads to a larger sample autocorrelation. The self-similarity of a time series computed using different time-lags is also referred to as serial correlation.

2.2.2 Predictability and Surprise

Intuitive notions from information theory, such as *information content* or *redundancy* [SW49], have been used to analyze certain characteristics of music such as pitch or temporal features [Mey57, You58, Coh62]. Viewing a composition as a particular selection of sounds forming some pattern in time [Coh62] paves the way for the application of information theoretic tools to music. Information can be intuitively described in the words of Weaver [Wea49b]:

> Information is ... a measure of one's freedom of choice in selecting a message. The greater this freedom of choice, and hence the greater the information, the greater is the uncertainty that the message actually is some particular one. Thus greater freedom of choice, greater uncertainty, greater information go hand in hand.

However, he later [Wea49a] adds, "... the word information relates not so much to what you *do* say, as to what you *could* say." In other words, although a Spartan view of music may reduce compositions to mere orderings of sounds, the artistic "value" of a composition comes from a composer's most critical task: selection of such sounds [Coh62]. Stravinsky defines such a selection as, "the need we feel to bring order out of chaos. ... To proceed by elimination — to know how to *discard*, as the gambler says, that is the great technique of selection" [Str70]. Nonetheless, to measure structural order of compositions using redundancy, Meyer notes [Mey56]:

> Styles in music are basically complex systems of probability relations in which the meaning of any term or series of terms depends upon its relationships with all other terms possible within the style system.

Cohen further extends this treatment of music as a sequence generated from a random source: "not only may the musical score be characterized as a probability system, but the listener *perceives* the music ... as a probability system." [Coh62] In other words, the tension between a composition's structural redundancy and unpredictability create its perceptual identity for the listener. Hindemith describes the "listener's act of perception" as [Hin52]:

> While listening to the musical structure, as it unfolds before his ears, he is mentally constructing parallel to it. ... Registering the composition's components as they reach him he tries to match them with their corresponding parts of his mental construction.

Hence, understanding structural patterns, built from the relationship between successive events, is a significant characterizer. Redundancy, as a quantitative measure of order, has been studied as an identifier for a body of music [Den18, You58]. In this thesis, I study the absence of order in inter-onset durations by measuring unpredictability through the *surprise index* (Definition 2.2.4). This index depends on the occurrence of unlikely events.

Definition 2.2.3 (Frequency Moments)**.** *Denote an alphabet, containing all possible values, by* $\Sigma_d = \{x_1 \dots x_n\}$. *Let* $\{d_1, \dots, d_m\}$ *be a sequence where* $d_i \in \Sigma_d, \forall 1 \leq i \leq m$, *and let* m_i *be the count of each alphabet element* d_i *in the sequence. That is,* $m_i = \{j | d_j == x_i\}$. *The kth frequency moment is defined as* $F_k = \Sigma_{i=1}^{n} m_i^k$.

It is assumed that standardized notation durations form a finite, small set of values. That is, although the durations calculated are not necessarily integers, the range of values that the duration can have in the music time series, composed by humans for humans, will be finite and small. I use four frequency moments in my analysis: F_0, F_1, F_2, F_∞. F_0 is the count of unique elements in the series, F_1 is the sum of unique values in the series, and F_2, also known as the *repeat rate* or "Gini's Index of homogeneity" [Goo89], is used to calculate the surprise index (Definition 2.2.4) [Wea48]. F_∞ is defined as the maximal frequency count of the alphabet: $\max_{1 \leq i \leq n} m_i$.

Definition 2.2.4 (Surprise Index [Wea48, Goo89])**.** *Denote the alphabet containing all possible values for a time series by* $\Sigma = \{x_1 \dots x_n\}$, *with corresponding probabilities*

$p_1 \ldots p_n$. *The Weaver surprise index,* $S_n(i)$, *is the ratio of the expected value of the probabilities to the observed probability* p_i*:* $S_n(i) = \frac{\Sigma_{j=1}^{n} p_j^2}{p_i}$.

The surprise index computes the ratio of the likelihood that an event is expected to occur to its actual occurrence [Wea48, Goo89]. Events that are unlikely - have a small probability of occurrence will generate a larger surprise index.

I compute surprise indices for the generated music duration time series in this analysis. The surprise index is calculated for all unique values in a time series and depends on its frequency of occurrence (i.e. *popularity*) in the series. The occurrence of an element that is not frequent is taken to be surprising, and hence should increase the local unpredictability of the series at that point in time. The surprise index, $S_n(i)$[1], is a global measurement for a music time series, in that its value depends on the global frequencies of unique elements of the series. This can be used to identify when least or most frequent elements in the time series occur. Identifying such key changing points can be used as a global indicator of surprise. In the present context, the surprise index will be used to determine compositions that are most surprising or atypical.

2.2.3 Measures of Variation

A time series that does not change at all represents an extreme: it has maximal redundancy and minimal variance. Such an invariant time series does not convey much information, and can be described concisely. Other time series have some degree of embedded variation and unpredictability. The following measures are used to quantify this variation.

[1] When the underlying alphabet and n are clearly understood, I simplify the notation and use S_i to denote $S_n(i)$.

Definition 2.2.5 (Coefficient of Variation)**.** *The coefficient of variation of time series $X(n)$ is defined as the ratio between the variance and the mean:*
$\texttt{coefficient of variation} = \frac{\sqrt{E[X^2(n)]-E[X(n)]^2}}{E[X(n)]}$

Variation can also be computed by measuring the degree of deviation of the values from the mean; i.e., the mean absolute deviation of the time series $X = \{x_1, \ldots, x_n\}$ is computed as the expected value of the deviation from the mean for each x_i: $E[x_i - E[X]]$.

In addition to computing the standard deviation, the interquartile range (Definition 2.2.6) is used to measure the spread of the data.

Definition 2.2.6 (Interquartile Range (IQR))**.** *The interquartile range (IQR) denotes the difference between the $75^{\texttt{th}}$ upper and the $25^{\texttt{th}}$ lower percentiles of the sample data. The upper and lower percentiles are computed by dividing the data into two parts at the median, computing the median for each of these newly formed portions, and then computing the medians of these portions: $Q1$ (lower quartile) and $Q3$ (upper quartile). The IQR is defined as $Q3 - Q1$.*

The IQR, mean and standard deviation may be used to test the normality of the underlying distribution generating the sample time series. However, IQR is robust to non-normal data distributions, while the standard deviation can be skewed by such data.

2.3 Emergence of Power Laws in Noise and Music

> *"There is no noise, only sound"*
>
> - John Cage [Koz92]

The study of power laws in music dates back to Zipf [Zip49], who examined the length of intervals between repetitions of notes and the number of melodic intervals on a very small set of music pieces. Mandelbrot, the father of fractal geometry [Man89], coined the term *fractional noise* to refer to fluctuations whose sample spectral density is of the form f^{1-2H}, where f denotes the frequency and H falls in $(0.5, 1)$ [Man67, MVN68]. He proposed this term in place of "1:f noise" since values of H both close to and far away from 1 are common [MVN68]. His research provided an alternative to the archetypical analysis of random functions through the lens of independence. He argued that empirical studies suggest an interdependence between distance samples. He highlighted a particular class of phenomena that were shown by Hurst [Hur51, HBS65] to exhibit such properties. Hurst, during his hydrological studies of the Nile, found that the accumulated water flows approximately varied according to $t^H, 1/2 < H < 1$ [Hur51, HBS65]. This significant empirical finding was further analyzed by Mandelbrot [MW69].

White, brown and pink noise have characteristics "expected to be 'typical' of what happens in the absence of asymptotic independence." [MVN68] These stochastic signals, termed "scaled noise" by Mandelbrot, exhibit spectral densities proportional to f^β, where $\beta \in (-2, 0)$ [Man67, MVN68]. In other words, similar to other fractional noise, they exhibit self-similarity in that they can be characterized through their features which are invariant with respect to time scale (scale-free noise) [MVN68].

White noise is an uncorrelated, stochastic time series: its autocorrelation — except at lag zero — is zero (Figure 2.1), and is characterized by its flat power spectra: $S(f) \propto (1/f)^0$ [Fuc03].

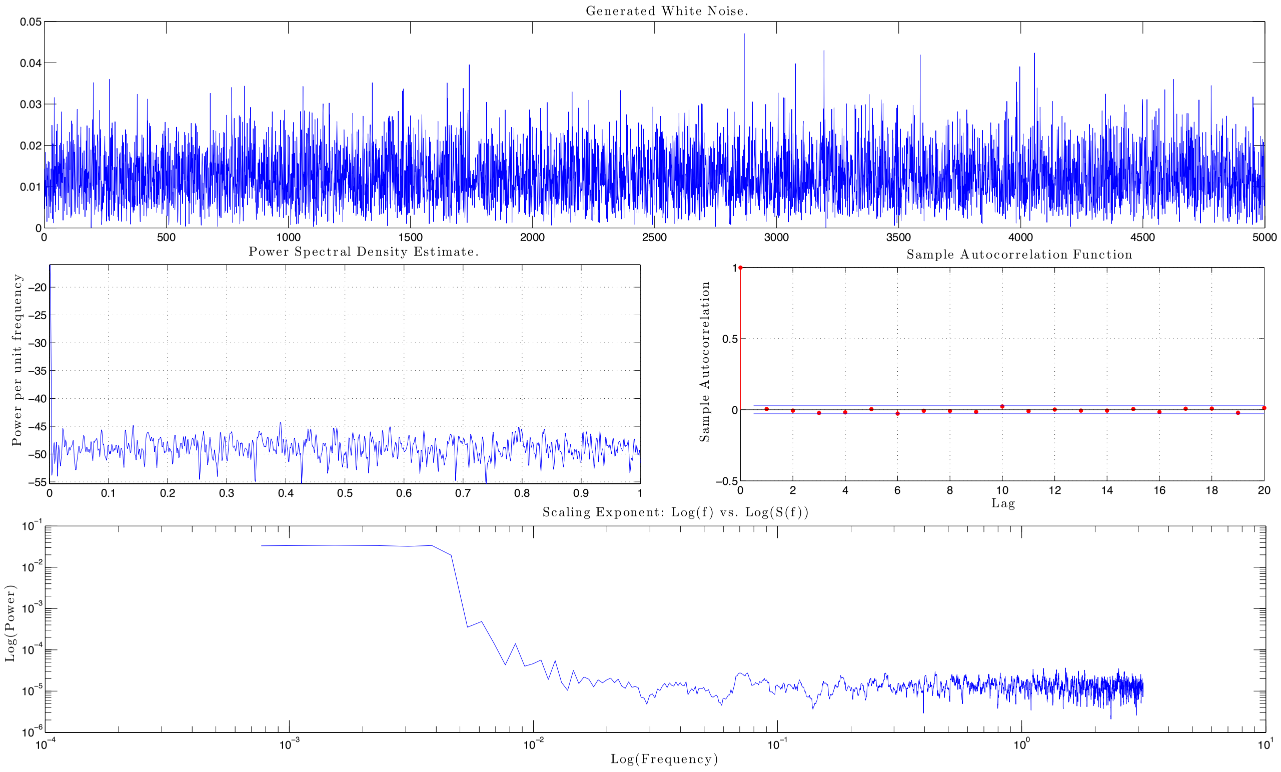


Figure 2.1: A sample white noise of length 5000 generated as an uncorrelated random signal. Its autocorrelation function, which is zero except at lag 1, and scaling exponent, $\beta = 0$, are displayed.

Filtered white noise is referred to as correlated or coloured noises (e.g., brown or pink noise) [Fuc03]. For instance, brown noise — named for its similarities to Brownian motion — is a scaling noise wherein each sample is influenced by its immediate history [Gar78]. It is a stochastic signal whose power spectral density decays as $S(f) \propto (1/f)^{-2}$ [MVN68, Fuc03] (Figure 2.2).

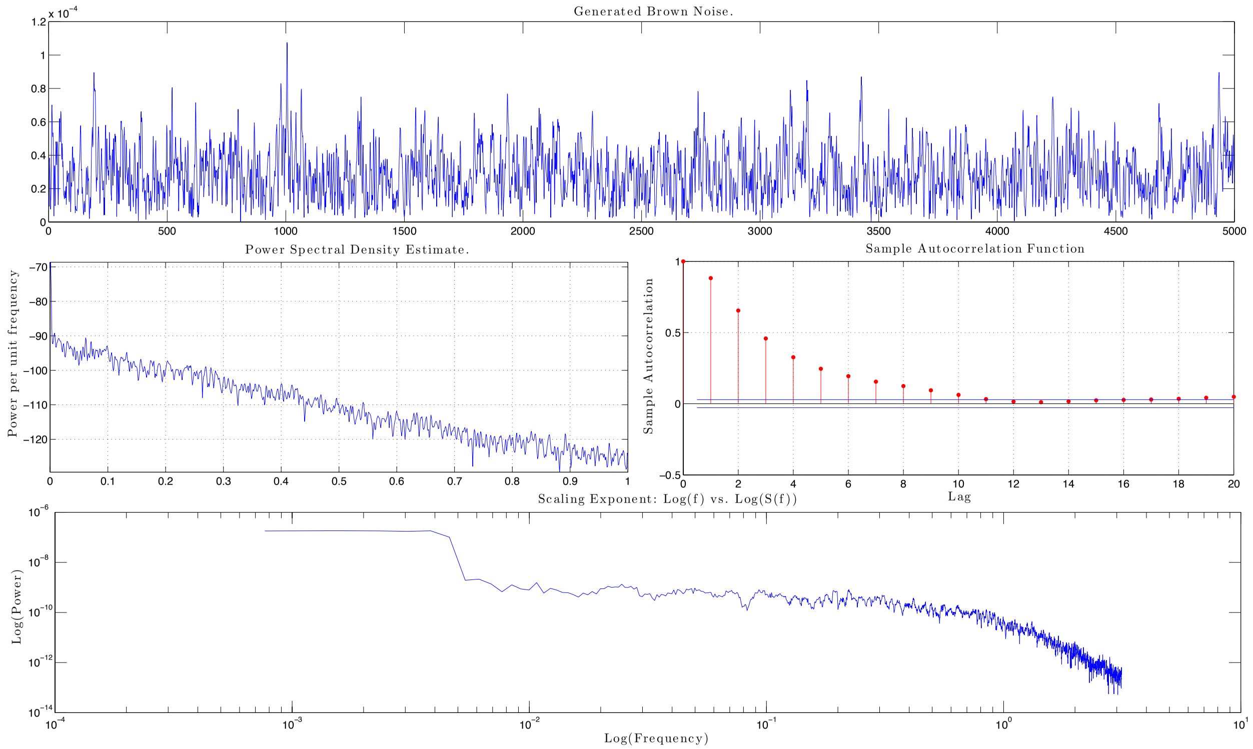


Figure 2.2: A sample brown noise of length 5000. Its autocorrelation function and power spectrum are displayed. Pink noise has spectral scaling exponent $\beta = -2$.

Lastly, many significant natural signals have scaling characteristics similar to those of fractal noise, also referred to as pink noise and which is characterized by $\beta = -1$ [Fuc03]. Such signals have spectral scaling exponents $\beta \approx -1$ [Gar78, Man83, Sch09], and will be further discussed in Chapter 3.

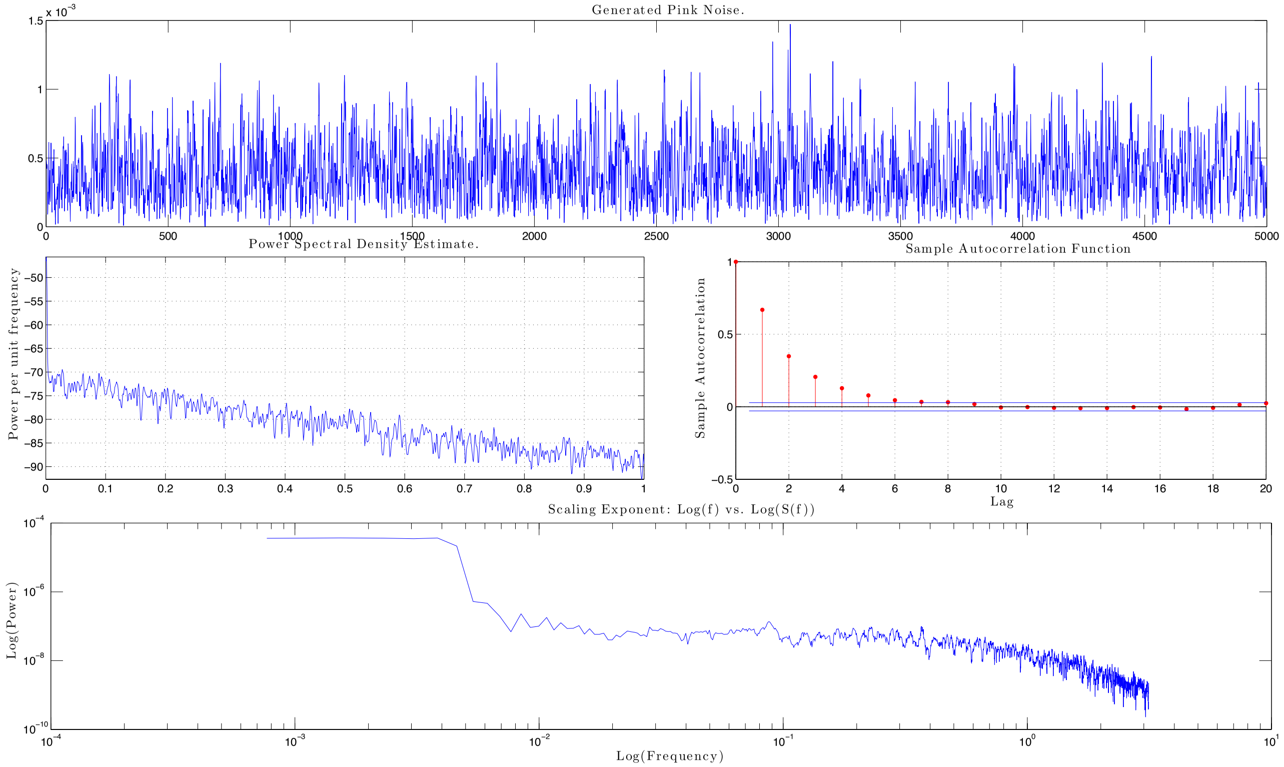


Figure 2.3: A sample brown noise of length 5000. Its autocorrelation function and power spectrum are displayed. Pink noise has spectral scaling exponent $\beta = -1$.

A comparison between the autocorrelation of pink noise (Figure 2.3) with brown (Figure 2.2) or white (Figure 2.1) noise, indicates an increase in correlation from white, to pink to brown noise [VC78, Gar78]. In the context of music, fractal noise has been shown to exist in pitch and loudness fluctuations [VC75, VC78, GTM95, PB00], changes of acoustic frequency [HH90, HH91], and rhythm [LCM12]. It has also been used for computer-generated compositions that are judged to sound aesthetically pleasing [VC78, Gar78, MVW$^+$03, Sch87].

2.3.1 Significance of Power Laws in Music

"Music is imitating the characteristic way our world changes in time."

Richard F. Voss [Vos88]

Power laws in music were first studied by Zipf, who studied the length of intervals between repetitions of notes and the number of melodic intervals on a very small set of music pieces [Zip49]. Subsequently, Voss and Clarke [VC75, VC78] studied the power law (scaling) behaviour for pitch and loudness fluctuations, and noted that the frequency fluctuations in music have a spectral density at frequencies down to the inverse of the length of the piece. They tested audio power and frequency fluctuations on a few audio examples taken from various genres of jazz, classical, blues, rock and radio recordings, and boldly hypothesized that: "the ubiquity of noise [asserts that] music mimics the way the world changes with time" [VC75]. This work was followed by Gilden et al. [GTM95] and Patel and Balaban [PB00], showing that power laws can characterize the pitch structure in melodies. Manaris et al. [MVW$^+$03] studied scaling laws in pitch, duration and melodic intervals of a 220-piece corpus ranging from baroque, classical, romantic, jazz, rock and random music. The immediate question would be to test whether this behaviour extends to other music features, specifically those that directly contribute to the identity of the piece, such as rhythm. It is surprising that there is a gap of a few decades between research on studying fractal-noise regularity analysis on the pitch structure in music and that on rhythm. This question was addressed by Levitin et al. [LCM12] who studied rhythm of 1788 movements from 558 compositions spanning 400 years of Western classical music. Table 2.3.1 shows a brief summary of previous literature on power laws in music, which will form a basis for this thesis' contributions in Chapters 3 and 4. The outlined state-of-the-art analytical approaches use distinct representations for symbolic music. Not withstanding the manual analyses of sheet music used by Zipf [Zip49], two other musical representations were used: (1) the Musical Instrumental Digital Interface standard (MIDI) and (2) Humdrum **kern [Hur93]. The latter encodes core musical information (e.g., duration information) as ASCII characters, and is a predefined representation used by the musical anslysis software kit Humdrum [Hur94]. On the other hand, in the MIDI representation, musical actions are described instead of sounded. For instance, the encoding specifies what note is played at what time.

	Musical Feature	**Scope of Analysis**	**Dataset Size**	**Music Representation**
Zipf [Zip49]	melodic interval, length of intervals between note repetitions	Mozart, Chopin, Irving Berlin, Jerome Kern	5	Printed scores
Voss and Clark [VC75, VC78]	pitch, loudness fluctuation, voltage	classical, jazz, blues, and rock radio station	Several hours of recordings	Audio
Hsü and Hsü [HH91]	acoustic frequency fluctuations	J. S. Bach's Invention no. 1 in C Major, BWV 772	1	Digitized score
Manaris et al. [MVW$^+$03, MRM$^+$05]	pitch and duration of musicals events	Western classical composers 8 genres	28 196	MIDI MIDI
Zanette [Zan06]	pitch and duration	J.S. Bach, Mozart, Debussy, Schoenberg	4	MIDI
Levitin et al. [LCM12]	rhythm	Western classical compositions	558	Humdrum **kern

Table 2.1: A brief overview of existing literature on power law analysis of various music features. Two representations of symbolic music are common: Humdrum **kern and MIDI [Hur94].

The popularization of studying power laws in music by the works of Voss and Clarke [VC75, VC78, Vos88] led to much debate as to the origins and necessity of such self-similarity characteristics in music. Voss argues [Vos88]:

> Both music and $1/f$-noise are intermediate between randomness and predictability. Like fractal shapes there is something interesting on all (in this case, time) scales. Even the smallest phase reflects the whole.

Another significant question initiated by the same line of work was whether fractal characteristics affect the aesthetic quality of music, or can be used to predict which musical pieces will be most liked. Voss and Clarke [VC78] conducted listening experiments on generated “white, $1/f$, and $1/f^2$ music” and concluded that, “$1/f$ music was judged by most listeners to be far more interesting than either the white music (which was ‘too random’) or the scalelike $1/f^2$ music (which was ‘too correlated’).”

Manaris [MVW$^+$03, MRM$^+$05] argues that aesthetically-pleasing music requires the existence of a fractal structure. That is, the intricacies considered in generating pleasing music will result in a structure that will manifest this scaling exponent.[2] Accordingly, Gardner notes: [Gar78]

> The changing landscape of the world (or, to put it another way, the changing content of my total experience) seems to cluster around $1/f$ noise. It is certainly not entirely uncorrelated, like white noise, nor is it as strongly correlated as brown noise. From the cradle to the grave my brain is processing the fluctuating data that comes to it from its sensors. If I measure this noise at the peripheries of the nervous system (under the skin of the fingers), it tends, Mandelbrot says, to be white. The closer one gets to the brain, however, the closer the electrical fluctuations approaches $1/f$. The nervous system seems to act like a complex filtering device, screening out irrelevant elements and processing only the patterns of change that are useful for intelligent behaviour.

This observation about the human disposition to appreciate and create complex musical structures — which paradoxically are elegant and simple because of their scale-free self-similarity — has been further motivated by other researchers as well. For instance, Levitin

[2] It should be noted that, as highlighted by Manaris [MVW$^+$03, MRM$^+$05], the fractal structure found is a necessary condition though not sufficient.

et al. [LCM12] explain the finding of fractal power laws in rhythms ($1/f$ structure) in music as having an evolutionary/biological basis. Because $1/f$ structure is ubiquitous in nature, they argue, the human brain evolved in such a way as to represent this regularity, alongside other regularities of the physical world such as gravity, entropy, momentum, and the forward motion of time [She87]. Composers introduce $1/f$ structure into their music because they have internalized the power law as a property of nature, and they write music to reflect these natural constants [VC78, Gar78, LCM12]. Human brains are evolved to find power laws pleasing, whether in the fractal patterns of snowflakes and flowers or in time series such as music [Gar78, Man83, WS90, MRM$^+$05, Bea07, WLY09]. According to the strong form of this argument, humans cannot help but to write music that conforms to a power law when attempting to make aesthetically pleasing works [Ber71, Sch09, Bea07].

2.4 Extraction of Structural Identity

This section, describes the methodology used for analyses in Chapters 3 and 4. These extract an information-rich structural information from a simple rhythmic representation of music.The approach described here, will be applied to my Western and non-Western repertoires of music scores for:

1. **Temporal Representation:** generate an efficiently-computable representation of rhythm as a sequence of durations (standard musical note lengths) extracted from scores.

2. **Scale-Free Structural Features:** investigate a concise representation of structural repetitions in musical rhythm.

- confirm the existence of temporal fractal relations in Western scores,
- demonstrate the existence of such self-similarity in Non-Western symbolic music,
- highlight variation in values of such fractal exponents both among various (Western and non-Western) *composers* and compositions of a particular composer.

3. **Structural Identity:** use scale-free structural information in rhythm for classification.

2.4.1 Temporal Representation: Music Time Series

Levitin et al. denote a *raster representation* of music to be an ordered sequence of the intervals between successive note onsets, which are extracted from musical scores [LCM12]. To extract rasters from musical scores, those authors used scripts from the Humdrum toolkit [Hur93]. In my analysis, rhythm is represented as music time series: a time series of *durations* of music events, measured in standardized quarter note lengths. I apply a new computational music analysis toolkit, `music21` [CA10], to extract durations. This toolkit provides for a hierarchical and event-based representation of music, "music21 stream" [AC11], and it has been applied for analysis in large symbolic music collections [CAF11]. I use this tool, in place of Humdrum scripts used in [LCM12], to implement a scalable, uniform framework of analysis for both MIDI and Humdrum **kern files in my repertories. The use of this object-oriented Python toolkit makes future analysis, modifications or search of music scores based on fractal characteristics readily possible.

2.4.2 Treatment of Silences in the Temporal Representation

> *"Music is the space between the notes."*
>
> - Achille-Claude Debussy [Koo08]

A key question — turned into a parameter of the design — is the treatment of silences (rests) in musical streams. Composers give silences an equal significance to notes; they both influence and help shape the unique *identity* of a piece. The significant effect of silences in forming musical identity, allows one to assign silences the same degree of significance as notes in an analysis. John Cage's famous "4' 33" tacet for any inst/insts" (1952) — consisting of three movements with only silences [PKG] — takes this particular treatment of silences to an extreme. This composition ensures that a piece is perceived by the audience devoid of any possible influence from the performers, the conductor and to some extent even the composer.

In this thesis, notated durations are extracted from collections of symbolic (notated) musical compositions. For each voice in a piece, durations of notes *and* rests are generated, and merged together. For instance, in this analysis, the music time series corresponding to the schematic excerpt shown in Figure 2.4 is:

$$m = [\underline{1.5, 0.25, 0.25, 0.5, 0.5}, \underline{1, 1, 1}, \underline{1.5, 0.25, 0.25, 0.5, 0.5}, \underline{1, 1, 1}, \underline{1.5, 0.25, 0.25, 0.5, 0.5}, \underline{2, 1}, \underline{1, 1, 0.75, 0.25}, \underline{0.5, 0.5, 0.5, 0.5, 1}].$$

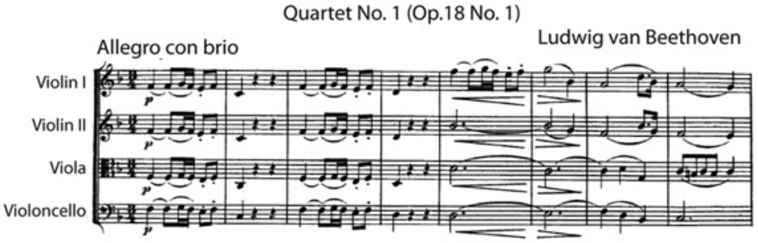


Figure 2.4: First eight measures from Beethoven's String Quartet No. 1 in F Major (Op. 18, No. 1).

The merged durations of notes and rests form a concise representation of the rhythmic content of a piece. By using notated compositions in this analysis, the set of possible durations in the music time series will be finite — in contrast to *performed* durations. For instance, Figure 2.5 visualizes the music time series corresponding to the first movement of Beethoven's Quartet No. 1 in F Major (Op. 18, No. 1), $r^{\texttt{Beethoven}}_{q_{18-1}}$. The durations in $r^{\texttt{Beethoven}}_{q_{18-1}}$ are measured in standardized quarter note lengths, and various voices (e.g., Violin I and Viola) are merged together. There are a total of 1948 durations in this music time series, $|r^{\texttt{Beethoven}}_{q_{18-1}}| = 1948$.

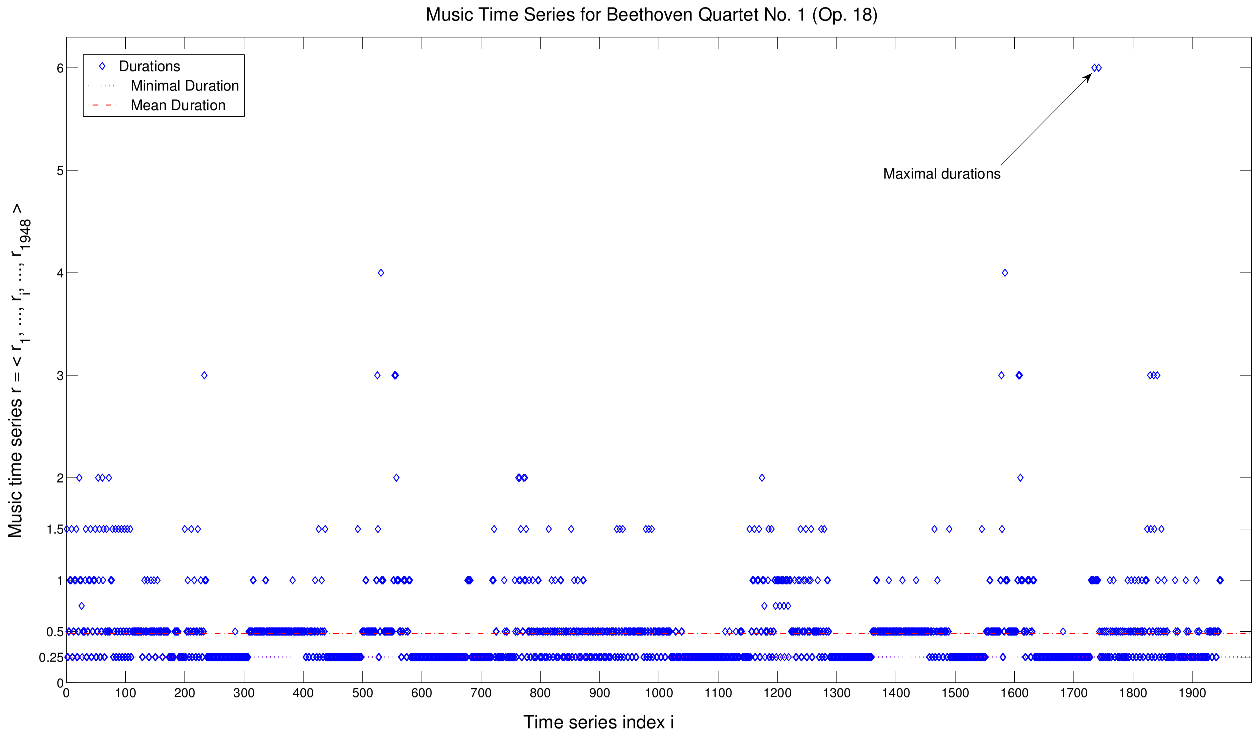


Figure 2.5: Music time series, $r^{\texttt{Beethoven}}_{q_{18-1}}$, corresponding to the first movement of Beethoven's Quartet Op. 18 No. 1 in F Major (Op. 18, No. 1).

As an example, some of the predictability features (Section 2.2) computed for the music time series corresponding to this Quartet are shown in Table 2.2. This shows that, the shortest duration is a sixteenth note here, and that there are only nine unique duration values in this music time series ($F_0 = 9$ as shown in Table 2.2).

Music Time Series	$r_{q_{18-1}}^{\mathtt{Beethoven}} = \langle r_1, \ldots, r_n \rangle$
Basic Statistics	$n = \|r\| = 1948$ $E(r_{q_{18-1}}^{\mathtt{Beethoven}}) = 0.48$ $\sigma(r) = 0.42$ $\mathtt{min}(\langle r_1, \ldots, r_n \rangle) = 0.25$ $\mathtt{max}(\langle r_1, \ldots, r_n \rangle) = 6$ Coefficient of variation$(R) = 0.87$ $E[r_i - E[r]] = 0.23$ IQR$(r) = 0.25$
Predictability Exponents (Π)	$\mathtt{SI}^{\mathtt{max}} = \mathtt{max(SI)} \;=\; 269.69$ $\|\{r_i \| \mathtt{SI}(r_i) = \mathtt{SI}^{\mathtt{max}}\}\| = 2$ $\mathtt{SI}\ \mathtt{SI}^{\mathtt{min}} = \mathtt{min}(\mathtt{SI}) = 0.55$ $\mathtt{SI}\ \|\{r_i \| \mathtt{SI}(r_i) = \mathtt{SI}^{\mathtt{min}}\}\| = 1$ $F_0 = 9$

Table 2.2: Basic statistical and predictability information about the first movement of Beethoven's Quartet No. 1 (Op. 18, No. 1).

2.4.3 Scale-Free Structural Features

I adapted the methods discussed by Levitin et al. [LCM12] to analyze features of rhythmic structure in my Western and non-Western symbolic music repertoires. Those authors focused on power law features in spectra of rhythm computed using a spectral estimation analysis, and refer to converging results computed using two time-domain analysis approaches. I briefly mention these methods in what follows. The analyses in Chapter 3 and 4 focus on analysis in the time domain to exhibit the manifestation of scale-free correlations in rhythmic structure of music. Classification in my analysis relies on the information content of musical rhythm (Table 2.4.4). This is quantified using information theoretic measures such as entropy and surprise index (*predictability exponents* - Section 2.2) and temporal power law correlation exponents (*fractal exponents*). The latter consist of scale-free exponents generated from analysis in the time domain (α, H) and spectral domain (β). These exponents are described below:

1. **Spectral Exponent (β_*):** Levitin et al. compute power law spectral exponent, β as the slope of a linear fit to the logarithm of (mean) power spectrum in the frequency range of 0.01 to 1 Hz. Though no particular justification is provided for the choice of frequency range, this is consistent with previous research. For instance, Voss and Clarke's reported fractal noise in music for a variable range of frequencies smaller than 1 Hz (as low as 0.002 Hz for a rock radio station and 4×10^{-4} Hz for a classical radio station) . For comparative analyses, what matters is that the same range is used for all time series under analysis. Thus, in my analyses, spectral exponents, $\beta_{\texttt{interpolate}}$, are estimated using parameters identical to those used in [LCM12].

Levitin et al. [LCM12] estimate the power spectra of the generated rhythmic rasters — their equivalent representation of music time series — using the Chronux toolbox [BAK^{+}10], which implements multi-taper spectral estimation. The authors compute the power law exponents as the slope of the linear regression fit to the log-log plot of frequency versus power spectra of rhythm, with only frequencies in the $0.01-1$ range considered. I denote these exponents by $\beta_{\texttt{interpolate}}$ in my analysis, and include two slightly different spectral exponents, denoted by $\beta_{\texttt{simpleFit}}$ and $\beta_{\texttt{mtm}}$. For each music time series, $\beta_{\texttt{simpleFit}}$ is computed identically to $\beta_{\texttt{interpolate}}$ with one difference: no frequency restriction in the linear fit. Lastly, $\beta_{\texttt{mtm}}$ exponents are computed using a different power spectral estimation toolbox; Matlab's built-in implementation of Thomson's multi-taper spectral estimation [Tho82, Mat] — without any frequency restriction — is used. A drawback of spectral analysis is an underlying assumption of stationarity, which may not hold for the majority of musical genres.

2. **Hurst Exponent (H):** Hurst analysis is an empirical method used to study long-range correlations in a time series. The Hurst exponent was first observed empirically by H. E. Hurst [Hur51] during his hydrological studies for the Nile, and was further analyzed by Mandelbrot [MW69]. There are various estimators for this exponent. In my analysis, I compute the Hurst exponent, H, of a music time series using a canonical rescaled range approach (*R/S estimation*) described in Algorithm 2.1. This exponent of self-similarity is estimated as $(R/S) \approx cn^{H}$: compute the rescaled range, and compute the slope of a linear regression fit to $\log(R/S)_n$ vs. $\log(n)$. H is a measure of long-range dependence in a time series: estimated as

0.5 for a random (uncorrelated) process [Hur51, HBS65]. Values greater than 0.5 are indicative of persistence (increases are likely to be followed by increases, and decreases with decreases) in a time series, and values smaller than 0.5 are indicative of anti-persistence processes. The Hurst exponent estimation with R/S has been shown to be sensitive to bias for short time series [DRL$^+$06].

3. **Zipf Exponent:** This power law exponent is the exponent of frequency's exponent decay with rank (most frequent events are not important) [Zip49]. This exponent, denoted by `simpleZipf` in my analysis, is computed through a linear regression applied to a log-transformed rank-frequency function of the data [WEG08]. In my discrete context, a histogram of duration values is computed, frequency equal to the number of elements contained in each bin is assigned to each bin (linear binning [WEG08]). Ranks are assigned in decreasing order: the bin with the most number of elements is assigned rank 1, and so forth. The exponent `simpleZipf` is the resulting slope of a linear regression fit to the logarithms of the ranks.

4. **Detrended Fluctuation Analysis - DFA Exponent (α):** Detrended Fluctuation analysis (DFA) introduced by Peng et al. [PHSG95] is an alternative approach for measuring long-range correlations for both stationary and non-stationary time series. The DFA analysis — a modified random walk analysis — studies fluctuations at various scales in a given time series without assuming any particular underlying characteristics such as stationaritiy [PHSG95, BS12]. In this approach, displayed in Algorithm 2.2, the input time series is first chopped into blocks of equal length n, and for each block, a local linear trend — a linear least-square fit to the block — is computed. For each local trend, variance of the difference between the original

Algorithm 2.1 Hurst Exponent Estimation

```
Require: Input raster $r = \{x_i\}_{i=1}^{L}$, where $|r| = L$ and $x_i \geq 0$.
while $L \geq 8$ do
    Divide $r$ into $d$ chunks of length $L$: $X^d = \{\{Z_{i,m}\}_{i=1}^{L}\}_{m=1}^{d} = \{Z_{1,m}, \ldots, Z_{L,m}\}_{m=1}^{d}$
    Compute Average $H$ Estimate: $(\frac{R}{S})_L \leftarrow$ avgHEstimate($X^d, L$)
    Store $(L, (\frac{R}{S})_L)$
    Divide: $L \leftarrow \frac{L}{2}$
end while
$(\texttt{slope}, \texttt{intercept}) \leftarrow \texttt{linearRegression}(\{(\log d, \log (\frac{R}{S})_d)\}_d)$.
$H \leftarrow \texttt{slope}$
return $H$

function avgEstimate($r, l$)
    Slice $r$ into portions $p_i$ of length $l$,
    for $i \in \{1, \ldots, d\}$ do,
        $\texttt{rescaledRange}_i \leftarrow \texttt{hEstimate}(p_i)$
    end for
    Compute the mean: $(\frac{R}{S})_l = \frac{1}{d}\Sigma_{i=1}^{d}\texttt{rescaledRange}_i$
return $(\frac{R}{S})_l$
end function

function hEstimate($p$)
    Compute the Mean: $E[p] \leftarrow \texttt{mean}(p)$
    Compute cumulative sum $y(t) \leftarrow \Sigma_{i=1}^{t} x_i - E[p]$
    Compute the Range: $R(n) \leftarrow \texttt{max}_t(y(t)) - \texttt{min}_t(y(t))$
    Compute the Standard Deviation: $S(n) \leftarrow \sigma(p)$
    Compute Rescaled Range: rescaledRange $\leftarrow \frac{R(n)}{S(n)}$
return $\texttt{rescaledRange}_n$
end function
```

values in the block and the corresponding estimates from the local trend is computed (*detrended walk*). The variance of this detrended walk is averaged over all blocks, denoted by $F^2(n)$. The DFA exponent, α, is the slope of a linear least-square fit to $\{\log F(n)\}_n$. The DFA analysis has been shown to be robust against oscillatory trends [KKBR+01, BS12], and it has been applied to a wide array of applications including correlation analysis in DNA [PBH+94, BGH+95], and stock market analyses[VAB97].

Algorithm 2.2 DETRENDED FLUCTUATION ANALYSIS EXPONENT

```
Require: Input raster $r = \{x_i\}_{i=1}^{L}$, where $|r| = L$ and $x_i \geq 0$. return DFA exponent, $\alpha$.
SEGMENT: $r = \{\{X_{i,m}\}_{i=1}^{n}\}_{m=1}^{d} = \{X_{1,m}, \ldots, X_{n,m}\}_{m=1}^{d}$
CUMULATIVE SUM SERIES: $\{Y_m\}_{m=1}^{d} = \{\{\Sigma_{t=1}^{i} X_{t,m}\}_{i=1}^{n}\}_{m=1}^{d}$
FOR $m \in \{1, \ldots, d\}$ DO
    $(\texttt{slope}_i, \texttt{intercept}_i) \leftarrow \texttt{linearRegression}(Y_m)$
    $F(m) = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_{i,m} - \texttt{slope}_m i - \texttt{intercept}_m)^2}$
END FOR
$\bar{F}(n) = \frac{1}{d}\Sigma_{m=1}^{d} F(m)$
$(\texttt{slope}, \texttt{intercept}) \leftarrow \texttt{linearRegression}(\log(n), \log(\bar{F}(n)))$
$\alpha \leftarrow \texttt{slope}$
```

2.4.4 Structural Identity

I use structural information in rhythm as discriminants to *classify* music in the repertoires by composer labels. That is, machine learning — a subfield of Artificial Intelligence (AI) [RNC+95] — algorithms are applied to use existing structural information in music to infer generalizations [Mit97]. In this context, I focus on supervised learning wherein information is extrapolated from *labelled* input examples (training dataset). In my analysis training datasets consist of the following features for each music time series:

- predictability features (Section 2.2.2),
- fractal exponents (Section 2.4), and
- composer labels (e.g., Mozart (Section 3.1) or Chinese music (Section 4.2.2))

Two supervised binary classifiers are trained on subsets of labelled datasets — Western (Section 3.1) and non-Western (Section 4.2.2) — and are evaluated on the remainder portions. That is, the goodness of these supervised classifiers is evaluated by measuring the fraction of correct composer label assignments for instances that were not included in the training phase; this process is repeated 10 times (10 fold cross-validation). Accuracy is measured as the fraction of instances correctly classified (i.e., assigned the correct composer label). In binary classifiers, with only two possible labels l_1 and l_2, accuracy is presented as a 2×2 matrix of correct classifications (l_i as $l_i, i = \{1, 2\}$) or misclassifications (l_i as l_j where $i \neq j, i, j \in \{1, 2\}$). In other words, accuracy is measured in an overall percentage of correct classification and presented in a *confusion matrix*. In the present context, a classifier uses the training set to assign *composer labels*. Table 2.3 summarizes the four possible outcomes of a confusion matrix for a binary classifier, with training set $R = \{p_j^{c_i}\}_{j=1}^{n}$, where $c_i \in \{0, 1\}$ denote the composer labels.

Table 2.3: A confusion matrix provides greater detail of accuracy. In the context of binary classifying composer labels c_0 or c_1. Classifying compositions of c_i as c_i, represents cross-diagonal entries in the matrix.

	$p_j^{c_0}$	$p_j^{c_1}$
$f(p_j^{c_*}) = c_0$	True positive	False positive
$f(p_j^{c_*}) = c_1$	False negative	True negative

My classification analysis applies logistic regression and decision trees, with 10-fold validation, to classify composer labels using features listed in Table 2.4.4. The two classifiers are implemented in WEKA [HFH$^+$09], a Java-based, open-source suite of machine learning algorithms, with no modifications. Classification in Chapters 3 and 4 use statistical features, denoted by Π, and fractal exponents (Table 2.4.4). In order to highlight the influence of fractal exponents, I distinguish between the following classification scenarios. Classification using

1. **Predictability and Fractal Exponents** - Denoted by $\{\Pi$, Fractal exponents$\}$, classification contains both sets of exponents

2. **Predictability Exponents** - Classification based on computed predictability features of Section 2.2.2, denoted by Π,

3. **Fractal Exponents** - Classification using only fractional exponents of Section 2.4, denoted by `Fractal Exponents`, and

4. **DFA Exponent** - Classification using only values of α in the training sets.

40

	Feature	**Label**

Fractal Exponents	Spectral exponents ($\beta_{\texttt{interpolate}}, \beta_{\texttt{mtm}}, \beta_{\texttt{simpleFit}}$)[LCM12]	`betaInterpolate`
		`betaSimpleFit`
		`betaMtm`

	Hurst exponent [Hur51] (Algorithm 2.1)	H
	DFA exponent [PHSG95] (Algorithm 2.2)	α
	Zipf exponent [Zip49, WEG08]	`simpleZipf`
Predictability Exponents (Π)	Maximal surprise index (`SI`)	$\texttt{SI}^{\texttt{max}} = \texttt{max(SI)}$
	Num of elements with this maximal `SI`	$\lvert\{r_i \lvert \texttt{SI}(r_i) = \texttt{SI}^{\texttt{max}}\}\rvert$
	Average temporal gap between elements of maximal `SI`	$E[G^{\texttt{max}}_{\texttt{si}}(t)]$
	Standard deviation of the temporal gap between elements with the maximal `SI`	$\sigma[G^{\texttt{max}}_{\texttt{si}}(t)]$
	Mean, medians and standard deviation of the duration values at these maximal surprise points	–
	Minimal surprise index `SI`	$\texttt{SI}^{\texttt{min}} = \texttt{min}(\texttt{SI})$
	Num of elements with this minimal `SI`	$\lvert\{r_i \lvert \texttt{SI}(r_i) = \texttt{SI}^{\texttt{min}}\}\rvert$
	Average temporal gap between elements of minimal `SI`	$E[G^{\texttt{min}}_{\texttt{si}}(t)]$
	Standard deviation of the temporal gap between elements with the maximal `SI`	$\sigma[G^{\texttt{min}}_{\texttt{si}}(t)]$
	Mean, medians and standard deviation of the duration values at these minimal surprise points	
	Zeroth frequency moment (F_0)	Definition 2.2.3
	First frequency moment (F_1)	Definition 2.2.3
	Second frequency moment (F_2)	Definition 2.2.3
	Maximal frequency moment (F_∞)	Definition 2.2.3
	Composer name	–

Table 2.4: List of features and their corresponding labels, as used in the proceeding figures and discussions, in classification.

In summary, the structural repetitions in music — quantified by temporal power law correlation features in rhythm — are computed for large repertoires of Western (Chapter 3) and non-Western (Chapter 4) symbolic music, and the informational significance of such temporal features as distinguishers between various composers are investigated through classification; This analysis approach is summarized in Figure 2.6.

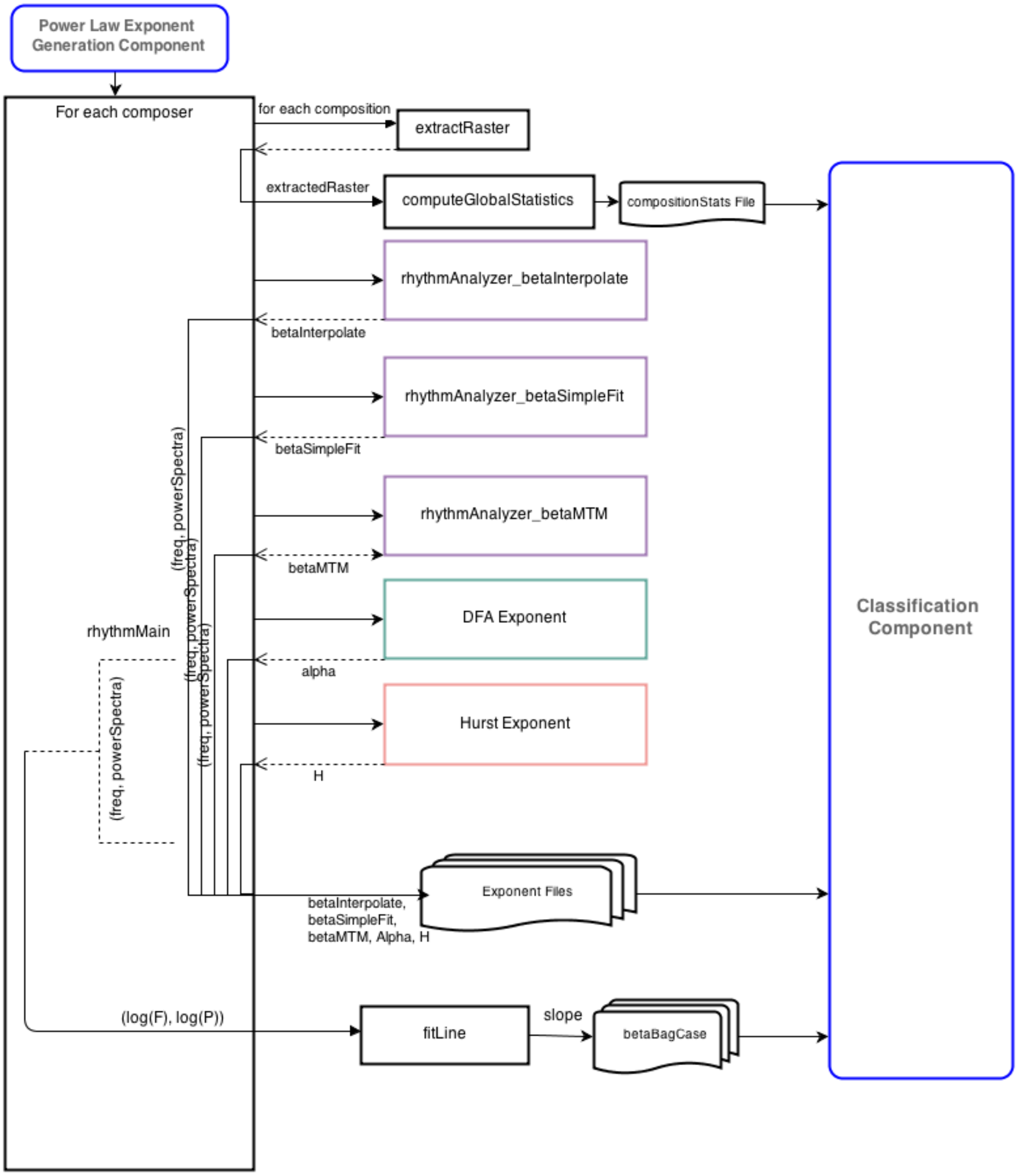


Figure 2.6: Two major components of analysis: power law exponent generation and classification. This figure focuses on the first and shows variation components generating the feature scaling exponents. The generated exponents are spectral exponent ($\beta_{\texttt{interpolate}}$, $\beta_{\texttt{simpleFit}}$, $\beta_{\texttt{mtm}}$), Hurst exponent (H), and DFA exponent (α). The exponents are computed for each composition of each composer under analysis.

3

Power Law Signatures for Western Compositions

> *"Now our job is to invent, or discover, a deep structure out of which that marvellous surface structure has been generated."*
>
> Leonard Bernstein [Ber76]

This chapter studies the scale-free structural information encoded in Western classical music rhythm. This is achieved by computing scale-free correlation exponents, *temporal fractal exponents*, and is further used to form a *composer signature*. The Western music collection consists of symbolic music only. This approach is used to analyze rhythmic power law features devoid of any performance idiosyncrasies. A composer-by-composer analysis of the computed fractal exponents is presented; variations and similarities for four particular composers are further discussed; and compositions of anomalous scale-free temporal exponents are highlighted. Finally, I present different cases of binary classifications to determine the significance of these global self-similarity exponents — representing the underlying power law phenomenon — as composer signatures.

3.1 Music Collection

The Western classical music collection used in my analysis consists of symbolic music taken from KernScores, an online library of musical scores [Sap05]. The collection consists of machine-readable music pieces, stored in the Humdrum **kern data format [Hur93], and includes compositions by a total of 24 Western composers from the $16^{\texttt{th}}$ to the $20^{\texttt{th}}$ Century (Figure 3.1). Composers included in my Western analyses lived in different eras, hailed from a range of distinct countries and had distinct musical styles (Table 3.3). The analyses presented in this chapter are based only on the music pieces included in this Western collection. Lastly, the choices of composers or their attributed music pieces are limited to those available, in the appropriate format, in the digital online music library at the time of this research.

Each music piece is represented as an a sequence of inter-onset durations of notated musical events, notes and silences, ordered in time. This is the concise temporal representation of rhythm, referred to as a music time series in this thesis (Section 2.4.1), chosen for the analysis of fractal exponents. The Western collection, $R^{\texttt{Western}}$, contains a total of 1165 music time series. Figure 3.2 shows a more granular statistical overview of the music series collection, $\{r_{c_j}\}_j$, corresponding to Western composer label, c_j.

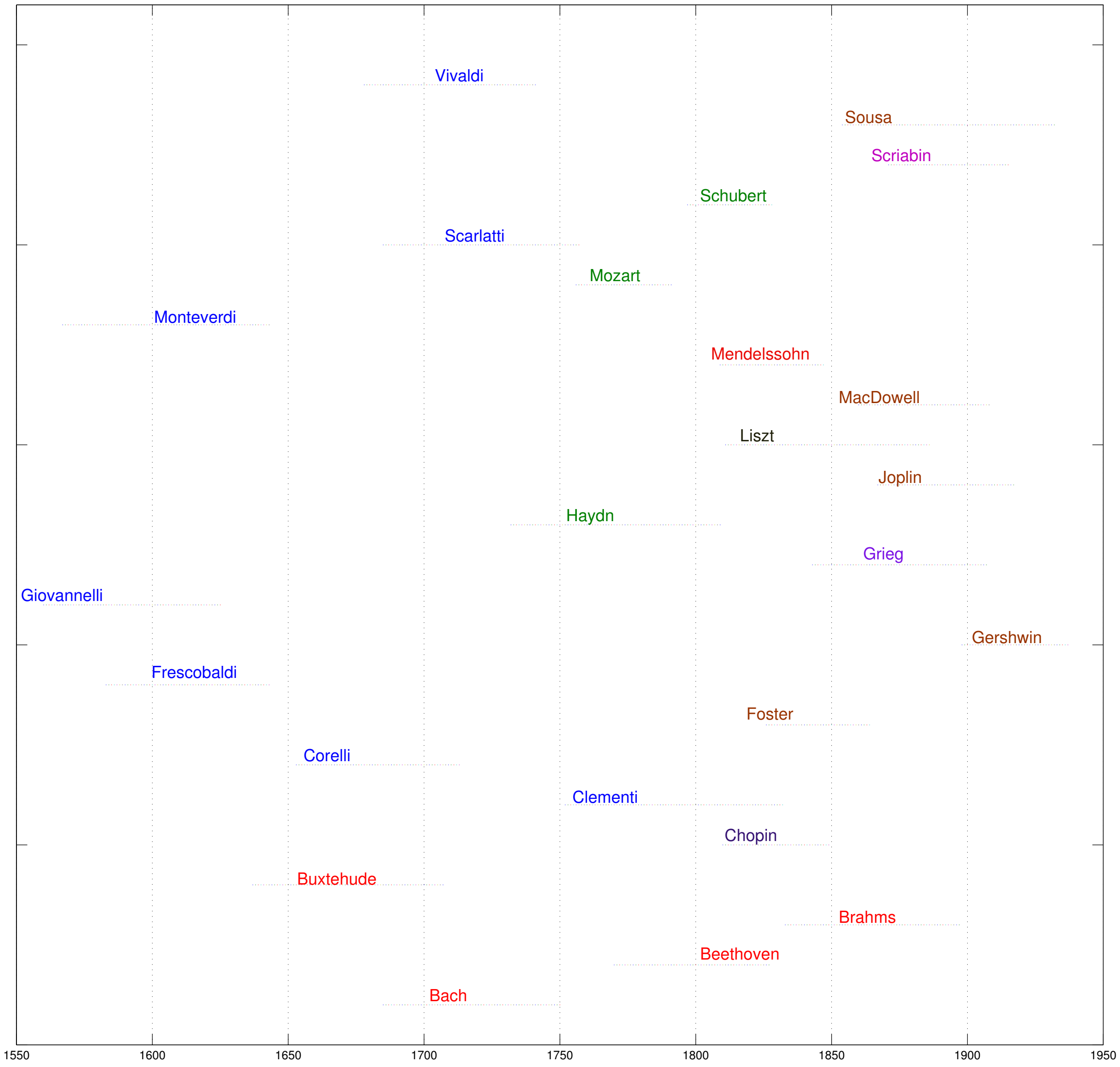
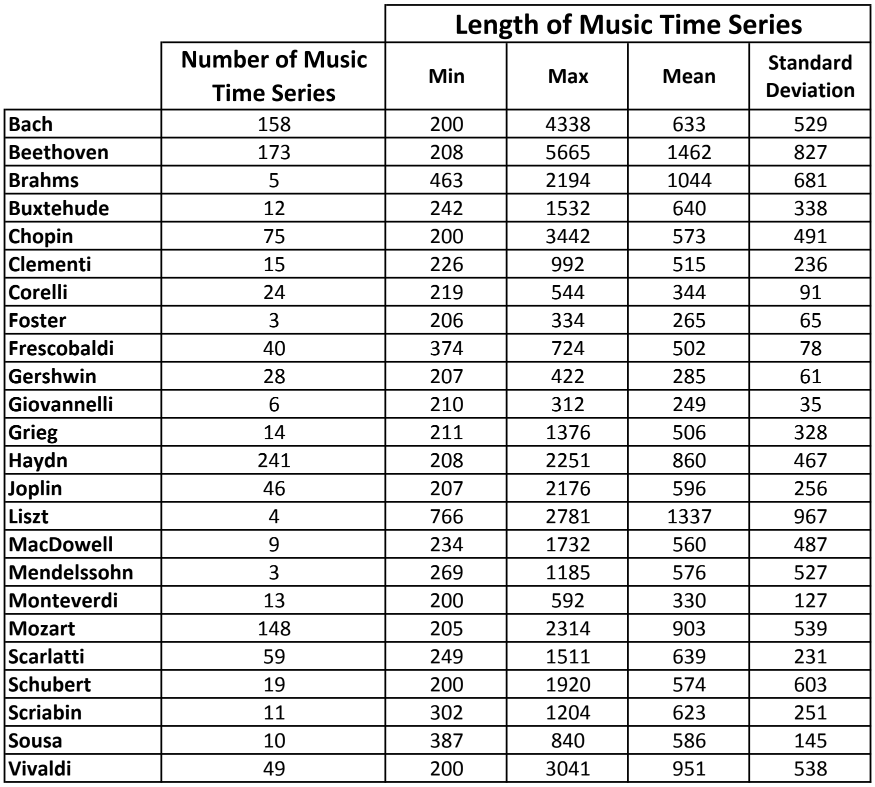


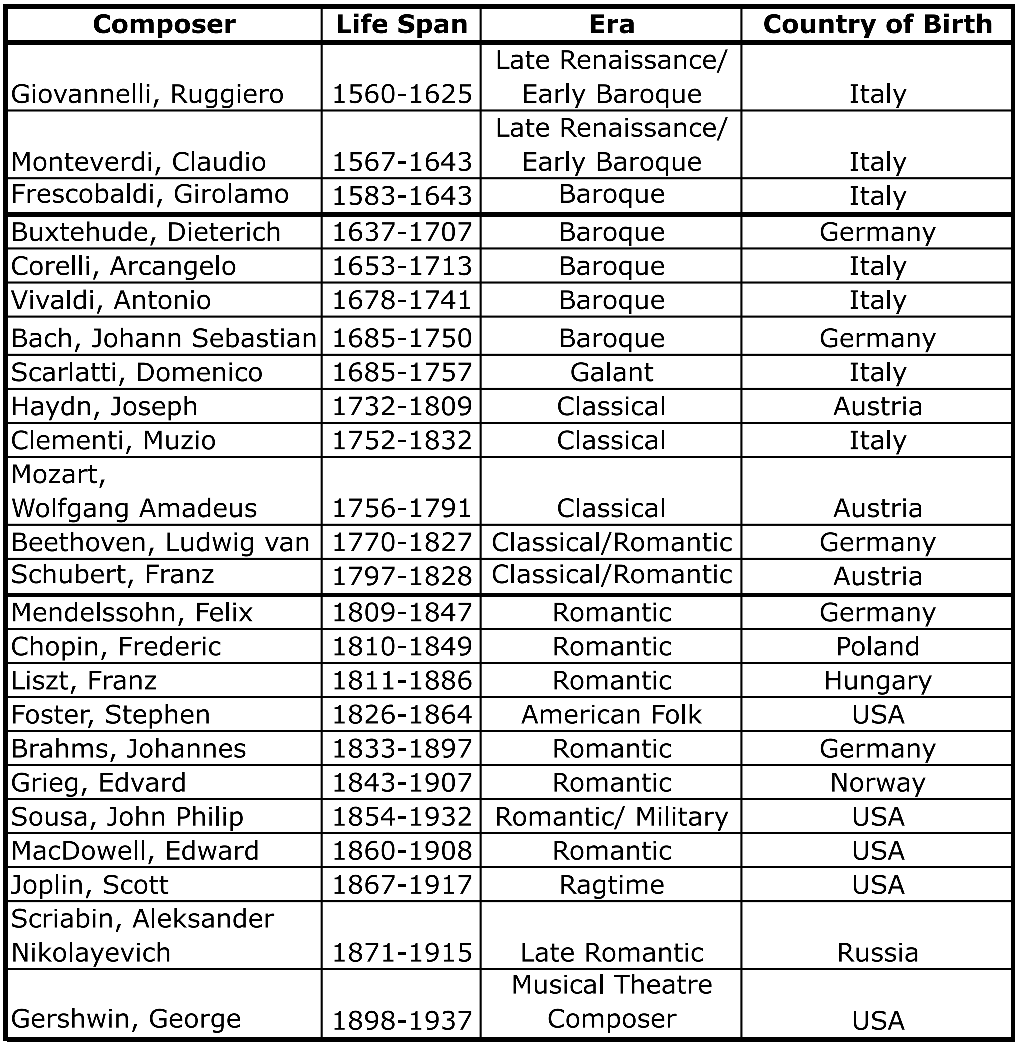
Figure 3.1: Chronological list of all Western composers. The plot's y-axis marks the number of composers (no unit). The x-axis represents time (measured in years). A composer's lifetime is represented by an interval, and composers who hailed from the same country are marked with the same colour.

The 24 composers analyzed here may have features such as era and country of origin in common (Figure 3.1). These similarities, in addition, to other distinctions are used to investigate the variability in fractal exponents of different composers, and are the basis of composer classifications. Depending on the availability of scores in KernScores at the time of this study, the number of music time series varies from composer to composer. The lengths of music time series, displayed in Figure 3.2, show variation from composer to composer. The number of music time series corresponding to Western classical composer c_j is denoted by $|R_{c_j}|$ Figure 3.2shows that each composer, c_j, has at least three music times series included in my analysis.



		Length of Music Time Series			
	Number of Music Time Series	Min	Max	Mean	Standard Deviation
Bach	158	200	4338	633	529
Beethoven	173	208	5665	1462	827
Brahms	5	463	2194	1044	681
Buxtehude	12	242	1532	640	338
Chopin	75	200	3442	573	491
Clementi	15	226	992	515	236
Corelli	24	219	544	344	91
Foster	3	206	334	265	65
Frescobaldi	40	374	724	502	78
Gershwin	28	207	422	285	61
Giovannelli	6	210	312	249	35
Grieg	14	211	1376	506	328
Haydn	241	208	2251	860	467
Joplin	46	207	2176	596	256
Liszt	4	766	2781	1337	967
MacDowell	9	234	1732	560	487
Mendelssohn	3	269	1185	576	527
Monteverdi	13	200	592	330	127
Mozart	148	205	2314	903	539
Scarlatti	59	249	1511	639	231
Schubert	19	200	1920	574	603
Scriabin	11	302	1204	623	251
Sousa	10	387	840	586	145
Vivaldi	49	200	3041	951	538

Figure 3.2: For each composer c_j in this Western collection, the number of music time series $|R_{c_j}| = |\{r_1^{c_j}, \ldots, r_n^{c_j}\}|$. Moreover, for each music time series in c_j's collection, the mean($E(|r_i^{c_j}|)$), sample standard deviation ($\sigma(|r_i^{c_j}|)$), $\mathtt{min}(|r_i^{c_j}|)$, and $\mathtt{max}(|r_i^{c_j}|)$ lengths of the music time series are computed. The composers are sorted alphabetically.



Composer	Life Span	Era	Country of Birth
Giovannelli, Ruggiero	1560-1625	Late Renaissance/ Early Baroque	Italy
Monteverdi, Claudio	1567-1643	Late Renaissance/ Early Baroque	Italy
Frescobaldi, Girolamo	1583-1643	Baroque	Italy
Buxtehude, Dieterich	1637-1707	Baroque	Germany
Corelli, Arcangelo	1653-1713	Baroque	Italy
Vivaldi, Antonio	1678-1741	Baroque	Italy
Bach, Johann Sebastian	1685-1750	Baroque	Germany
Scarlatti, Domenico	1685-1757	Galant	Italy
Haydn, Joseph	1732-1809	Classical	Austria
Clementi, Muzio	1752-1832	Classical	Italy
Mozart, Wolfgang Amadeus	1756-1791	Classical	Austria
Beethoven, Ludwig van	1770-1827	Classical/Romantic	Germany
Schubert, Franz	1797-1828	Classical/Romantic	Austria
Mendelssohn, Felix	1809-1847	Romantic	Germany
Chopin, Frederic	1810-1849	Romantic	Poland
Liszt, Franz	1811-1886	Romantic	Hungary
Foster, Stephen	1826-1864	American Folk	USA
Brahms, Johannes	1833-1897	Romantic	Germany
Grieg, Edvard	1843-1907	Romantic	Norway
Sousa, John Philip	1854-1932	Romantic/ Military	USA
MacDowell, Edward	1860-1908	Romantic	USA
Joplin, Scott	1867-1917	Ragtime	USA
Scriabin, Aleksander Nikolayevich	1871-1915	Late Romantic	Russia
Gershwin, George	1898-1937	Musical Theatre Composer	USA

Figure 3.3: Western classical composers included in the Western fractal signature analysis. Composers are sorted chronologically by year of birth, and further categorized by their corresponding musical era and last name.

In this collection of compositions, a simple count of unique values in all music time series reveals that the most frequently-used duration value is 1, corresponding to the notated duration of the musical quarter note, often associated by musicians with one beat (or the *tactus*). Also, a simple calculation reveals that, on average, the music time series contain more randomness (i.e., are more unpredictable) in their second half. That is, the (Shannon) entropy computed for a time series' first half is lower than its second half; there is a small increase in entropy from 1.69 ± 1.2 to 1.80 ± 1.3 ($\mu_H \pm \sigma_H$), though this is not statistically significant.

3.1.1 Limitations of the Music Collection

The choices of compositions analyzed here are not random. For the sake of consistency, I only included music pieces from the KernScores online library of musical scores [Sap05]. Although this collection provides for a great starting point, extending the analysis discussed in this thesis to a more comprehensive collection of Western music is left for future research. Moreover, this collection of available classical music scores contains music pieces for which the year of composition are not definitively known. In those cases, either the approximate year of composition — according to appropriate musical history records — or the average of the attributed compositional interval is used in my analysis.

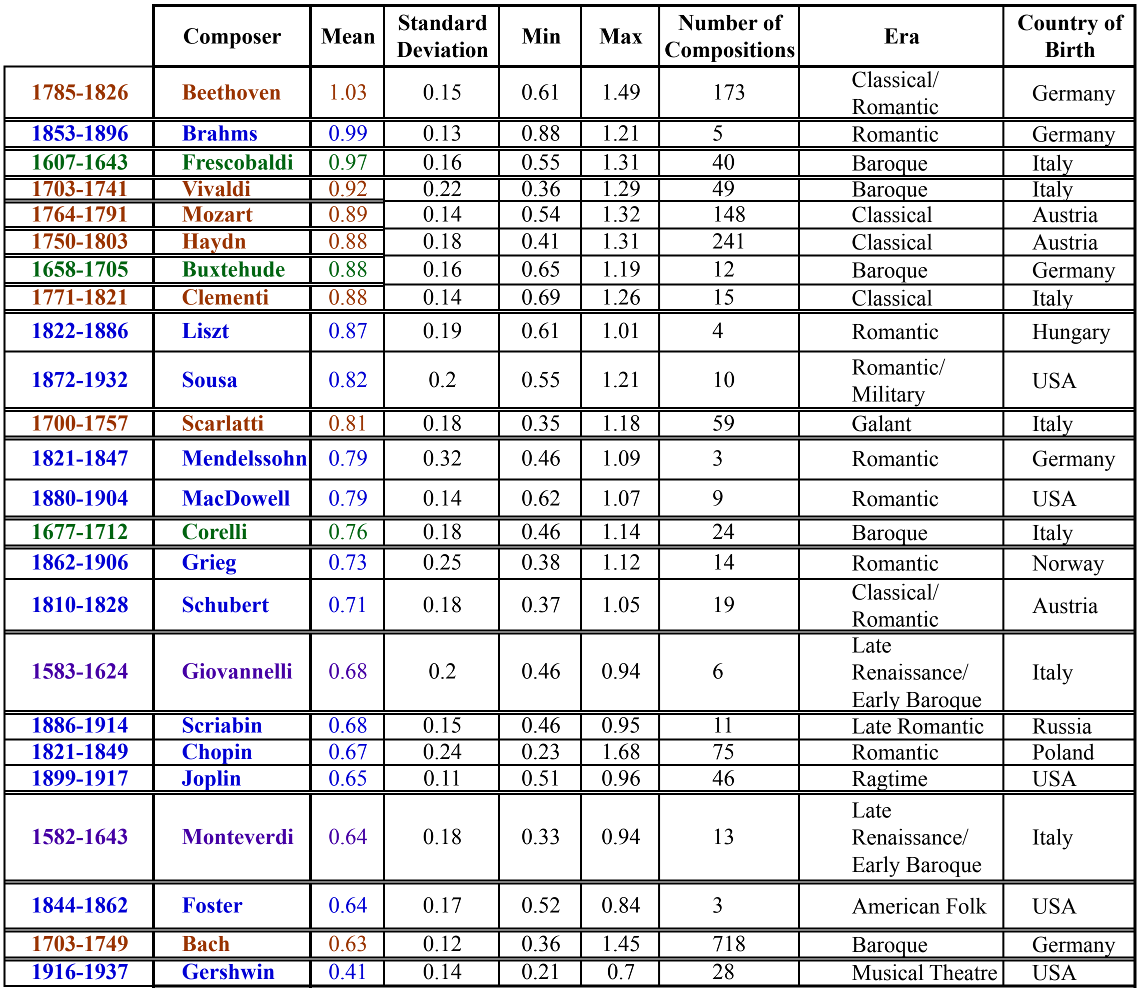
3.2 Western Temporal Power Law Spectral Exponents

Section 2.4 outlined methods used to generate scale-free temporal correlation exponents for music time series. DFA and Hurst fractal exponents, denoted by α and H, are computed using Algorithms 2.2 and 2.1 respectively. The average behaviour of these exponents are summarized for each Western composer , based on his corresponding sample music time series included here.

3.2.1 Detrended Fluctuation Analysis Scaling Exponents

DFA power law exponents, α, are computed using the Detrended Fluctuation Analysis algorithm [PHSG95] (Algorithm 2.2) for compositions available in this Western collection whose music time series have a minimum length of 200. This minimum length requirement — identical to the minimum length of rasters in [LCM12] — ensures that the duration of each music piece is sufficiently long to contain characteristic information; such a restriction will not eliminate may music time series in the Western collection analyzed here (Figure 3.2).

The DFA exponents of the Western collection analyzed in this dissertation are displayed in Figures 3.4, 3.6 and 3.5. The mean DFA exponents fell in the $0.5 - 1$ range (Figures 3.4 and 3.6); that is, fractal power law behaviour emerged in the music time series of the Western compositions studied. The α values, though similar for some composers, vary from composer to composer, and are distinct for different compositions of a particular composer (Figures 3.4).



	Composer	Mean	Standard Deviation	Min	Max	Number of Compositions	Era	Country of Birth
1785-1826	Beethoven	1.03	0.15	0.61	1.49	173	Classical/ Romantic	Germany
1853-1896	Brahms	0.99	0.13	0.88	1.21	5	Romantic	Germany
1607-1643	Frescobaldi	0.97	0.16	0.55	1.31	40	Baroque	Italy
1703-1741	Vivaldi	0.92	0.22	0.36	1.29	49	Baroque	Italy
1764-1791	Mozart	0.89	0.14	0.54	1.32	148	Classical	Austria
1750-1803	Haydn	0.88	0.18	0.41	1.31	241	Classical	Austria
1658-1705	Buxtehude	0.88	0.16	0.65	1.19	12	Baroque	Germany
1771-1821	Clementi	0.88	0.14	0.69	1.26	15	Classical	Italy
1822-1886	Liszt	0.87	0.19	0.61	1.01	4	Romantic	Hungary
1872-1932	Sousa	0.82	0.2	0.55	1.21	10	Romantic/ Military	USA
1700-1757	Scarlatti	0.81	0.18	0.35	1.18	59	Galant	Italy
1821-1847	Mendelssohn	0.79	0.32	0.46	1.09	3	Romantic	Germany
1880-1904	MacDowell	0.79	0.14	0.62	1.07	9	Romantic	USA
1677-1712	Corelli	0.76	0.18	0.46	1.14	24	Baroque	Italy
1862-1906	Grieg	0.73	0.25	0.38	1.12	14	Romantic	Norway
1810-1828	Schubert	0.71	0.18	0.37	1.05	19	Classical/ Romantic	Austria
1583-1624	Giovannelli	0.68	0.2	0.46	0.94	6	Late Renaissance/ Early Baroque	Italy
1886-1914	Scriabin	0.68	0.15	0.46	0.95	11	Late Romantic	Russia
1821-1849	Chopin	0.67	0.24	0.23	1.68	75	Romantic	Poland
1899-1917	Joplin	0.65	0.11	0.51	0.96	46	Ragtime	USA
1582-1643	Monteverdi	0.64	0.18	0.33	0.94	13	Late Renaissance/ Early Baroque	Italy
1844-1862	Foster	0.64	0.17	0.52	0.84	3	American Folk	USA
1703-1749	Bach	0.63	0.12	0.36	1.45	718	Baroque	Germany
1916-1937	Gershwin	0.41	0.14	0.21	0.7	28	Musical Theatre	USA

Figure 3.4: DFA (α) exponents for Western composers sorted in a decreasing order of mean α. The composers professional time lines are used to categorize the composers in to groups with distinct colors: (1) Early to mid 17^{th} century, (2) Mid to late 18^{th} century, (3) Early 20^{th} century.

In this collection of Western composers, Beethoven — with the highest mean $\alpha = 1.03$ — and Gershwin — with the lowest mean $\alpha = 0.41$ — represent the composers with the most and least predictable rhythmic structures, respectively (Figure 3.4). Gershwin's minimal DFA exponent, indicates that his rhythmic compositional style contains fewer long-range correlations. One of Gershwin's signatures is that he helps the listener learn the motifs that are to come by restating the opening material, and then he follows that with contrasting material [RC]. At the other end of the extreme, Figure 3.5 indicates that Beethoven's compositions , analyzed in this sample collection, are more structurally correlated, and contain more long-range patterns. It is interesting to note the similarity between the exponents of Beethoven and Brahms computed in my analysis, which may be attributed to similarities in compositional style of these composers [BW].

Figure 3.4 highlights similarities in the DFA exponents computed, for music pieces included in this Western collection. For instance, Haydn $(1750 - 1803)$ has a mean α of $0.88(\pm\ 0.18)$ and Liszt $(1822 - 1886)$ has $0.87(\pm\ 0.19)$, where the values inside the brackets represent the corresponding standard deviations.

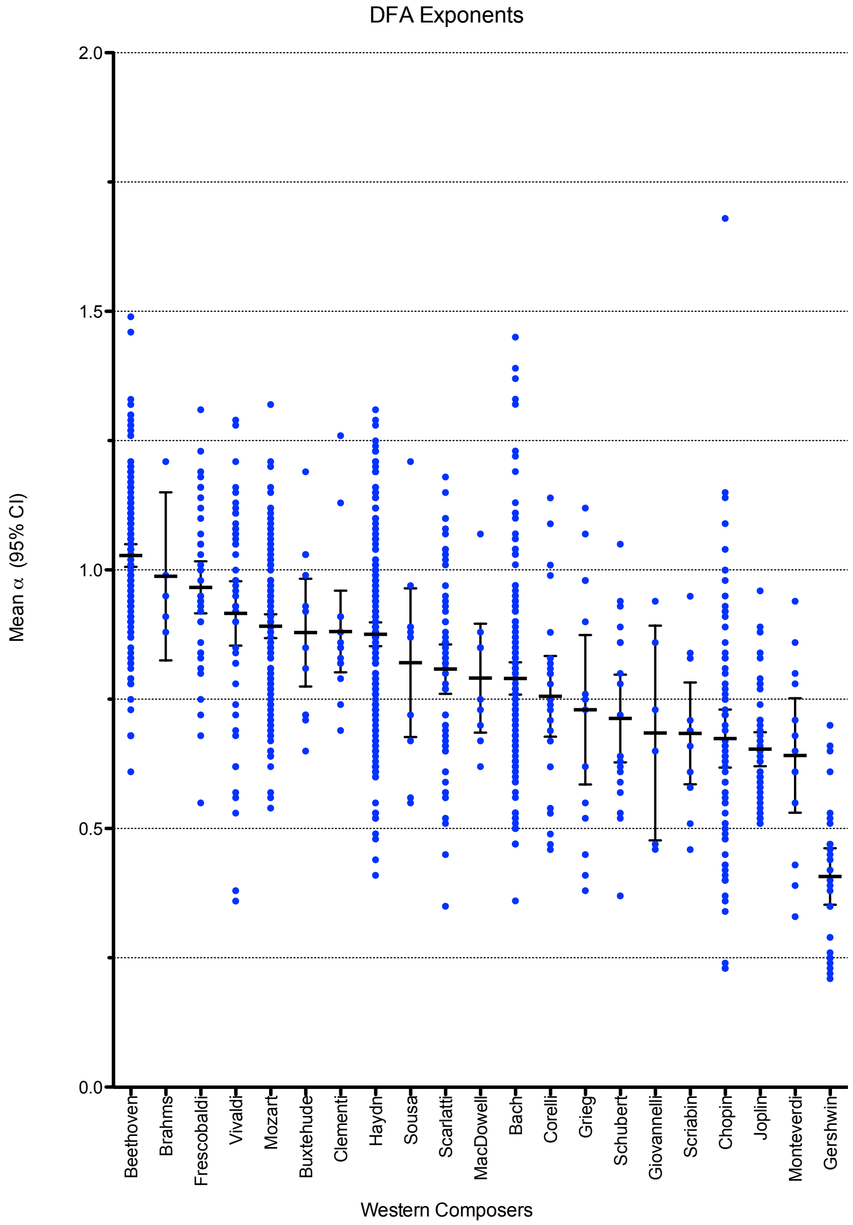


Figure 3.5: The Detrended Fluctuation Analysis (DFA) Exponents, α, computed for each music piece in the Western classical collection presented. The exponents are sorted in decreasing mean values. For each composer considered the mean 95% confidence interval is marked in black.

The values of α computed for composers in this Western collection are shown in Figure 3.5.

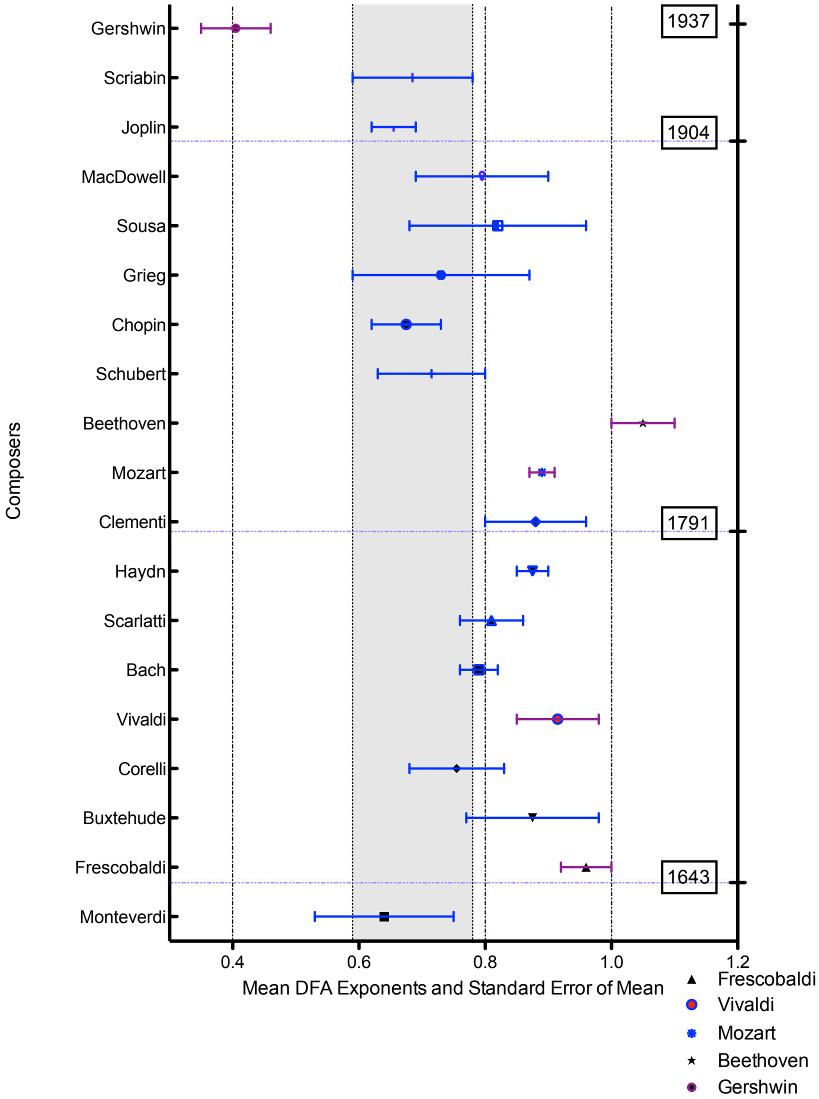


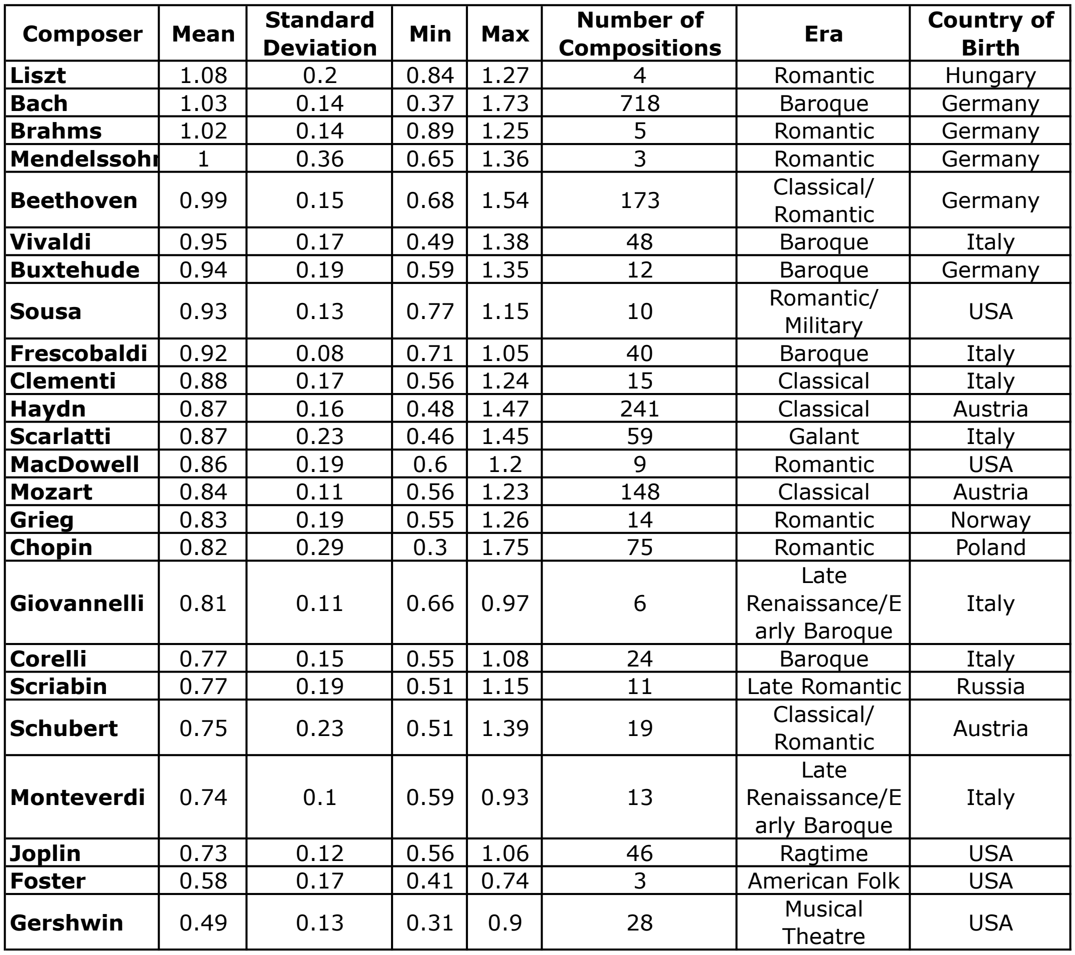
Figure 3.6: Mean DFA exponents. Each interval indicates the mean and the standard error mean for a particular Western composer in this study. Composers are sorted chronologically, and only those with 15 music time series in the Western collection of Section 3.1 are displayed.

The variability of the presented temporal fractal exponents — both for different composers as well as various compositions of a particular composer — highlight the richness of rhythmic structures in music. It is surprising that some of the most well known composers have highly concentrated α ranges. For instance, this study shows that the DFA exponents of music time series that were available for Bach, Mozart, and Scarlatti have small standard errors of mean (Figure 3.6).

3.2.2 Hurst Exponents

The Hurst exponent (H), measuring long-range correlations [Hur51, HBS65], is computed for each music time series using Algorithm 2.1 described in Section 2.4.

$H^{\texttt{Gershwin}}_{r_i}\_r_i \in R_{\texttt{Gershwin}}) \approx 0.5$ where $r_i \in R_{\texttt{Gershwin}}$, is notably smaller than other composers' (Figure 3.7). On the other hand, Beethoven's Hurst values ($E[H(R^{\mathsf{Beethoven}})] = \mu(\{H^{\texttt{Beethoven}}_{r_i}\}_{r_i \in R_{\texttt{Beethoven}}}) = 0.99$) — similar to his DFA exponents ($E[\alpha(R^{\mathsf{Beethoven}})] = \mu(\{\alpha^{\texttt{Beethoven}}_{r_i}\}_{r_i \in R_{\texttt{Beethoven}}}) = 1.03$) discussed in Section 3.2.1 — highlight the existence of highly structured patterns in his compositions available in my collection, and confirm Levitin et. al.,'s previous finding that Beethoven's rhythms are among the most predictable [LCM12]. Lastly, Frescobaldi's forty compositions analyzed here, exhibit the least amount of variation in H. It is evident that although there are similarities between the power law exponents of all composers, there is also considerable variation.



Composer	Mean	Standard Deviation	Min	Max	Number of Compositions	Era	Country of Birth
Liszt	1.08	0.2	0.84	1.27	4	Romantic	Hungary
Bach	1.03	0.14	0.37	1.73	718	Baroque	Germany
Brahms	1.02	0.14	0.89	1.25	5	Romantic	Germany
Mendelssohı	1	0.36	0.65	1.36	3	Romantic	Germany
Beethoven	0.99	0.15	0.68	1.54	173	Classical/ Romantic	Germany
Vivaldi	0.95	0.17	0.49	1.38	48	Baroque	Italy
Buxtehude	0.94	0.19	0.59	1.35	12	Baroque	Germany
Sousa	0.93	0.13	0.77	1.15	10	Romantic/ Military	USA
Frescobaldi	0.92	0.08	0.71	1.05	40	Baroque	Italy
Clementi	0.88	0.17	0.56	1.24	15	Classical	Italy
Haydn	0.87	0.16	0.48	1.47	241	Classical	Austria
Scarlatti	0.87	0.23	0.46	1.45	59	Galant	Italy
MacDowell	0.86	0.19	0.6	1.2	9	Romantic	USA
Mozart	0.84	0.11	0.56	1.23	148	Classical	Austria
Grieg	0.83	0.19	0.55	1.26	14	Romantic	Norway
Chopin	0.82	0.29	0.3	1.75	75	Romantic	Poland
Giovannelli	0.81	0.11	0.66	0.97	6	Late Renaissance/Early Baroque	Italy
Corelli	0.77	0.15	0.55	1.08	24	Baroque	Italy
Scriabin	0.77	0.19	0.51	1.15	11	Late Romantic	Russia
Schubert	0.75	0.23	0.51	1.39	19	Classical/ Romantic	Austria
Monteverdi	0.74	0.1	0.59	0.93	13	Late Renaissance/Early Baroque	Italy
Joplin	0.73	0.12	0.56	1.06	46	Ragtime	USA
Foster	0.58	0.17	0.41	0.74	3	American Folk	USA
Gershwin	0.49	0.13	0.31	0.9	28	Musical Theatre	USA

Figure 3.7: Mean Hurst values (H) for Western composers of Table 3.3. Rows are sorted in a descending order of H. Each row is highlighted by three colors: green for composers born in the $16^{\mathtt{th}}$, blue for those born in the $17-18^{\mathtt{th}}$ and red for the composers of $19-20^{\mathtt{th}}$ century.

For the sample Western collection available here, Section 3.3 presents presents further analysis for groups of composer — where there is a common characteristic amongst composers in each group.

3.3 Fractal Exponents: Influence of Time and Geography

A composer's compositions may be influenced both by the musical style of his era (time period factor), his contemporaries (competition factor), as well as his personal preferences molded by his upbringing (nationality factor). These factors are not, by any measure, an exhaustive list of all that may influence a composer's compositional style. Tables 3.1 and 3.2 demonstrate temporal power law exponents computed for composers grouped by time period and country of origin. The results outlined therein show that indeed composers with similar time period, competition and nationality factors may have structurally-similar compositions. These similarities are further discussed in the binary classification Section 3.5.

3.3.1 Similar Time Era

This section presents fractal exponents for composers that have lived during the same time period (i.e., they are coeval), and had similar corresponding periods of professional activity. The beginning of a professional timeline is marked by a composer's first composition, and unless otherwise stated, all proceeeding timelines correspond to periods of professional activity.

Composer (c_j)	Life Span	Professional Timeline	$\|R^{c_j}\|$	$E[\alpha]$	$E[H]$
Giovannelli, Ruggiero	1560 – 1625	1583 – 1624[1]	6	0.68	0.81
Monteverdi, Claudio	1567 – 1643	1582 – 1643[2]	13	0.64	0.74
Buxtehude, Dieterich	1637 – 1707	1658 – 1705[3]	21	0.88	0.94
Corelli, Arcangelo	1653 – 1713	1677 – 1712[4]	47	0.76	0.77
Bach, Johann Sebastian	1685 – 1750	1703 – 1749[5]	908	0.79	0.92
Scarlatti, Domenico	1685 – 1757	1700 – 1757[6]	59	0.81	0.87
Vivaldi, Antonio	1678 – 1741	1703 – 1739[7]	61	0.92	0.95
Joplin, Scott	1867 – 1917	1899 – 1917[8]	46	0.65	0.73
Sousa, John Philip	1854 – 1932	1872 – 1932[9]	10	0.82	0.93
Scriabin, Aleksander Nikolayevich	1871 – 1915	1886 – 1914[10]	13	0.69	0.77

Table 3.1: Western Composers' Fractal Exponents. Temporal power law exponents listed here are mean values of exponents computed using the DFA (Algorithm 2.2) and Hurst (Algorithm 2.1) algorithms. Total number of compositions analyzed for composer c_j is denoted by $|R^{c_j}|$.

Table 3.1 demonstrates similar exponents, based on the samples collection studied here. However, there are also coeval composers with distinct temporal fractal exponents, in my study. For instance, mean α value for Vivaldi $(1703 - 1739)$ is higher than that of Scarlatti $(1700 - 1757)$, in this Western collection. Also, my investigation found distinctions between the power law signatures of Bach and Vivaldi: That is, Figure 3.6 shows a distinction between the mean and standard error of mean of the DFA Exponents computed for the Bach $(1703 - 1749)$ and Vivaldi $(1703 - 1741)$ sub-categories available for analysis here.

3.3.2 Similar Origins

Patel [Pat06] shows that the native language of a composer and the structure of his compositions may be correlated. I investigate this influence for four composers born in Germany. Here, I focus on Western composers hailing from the same country as an indicator of a commonality between their first languages (mother tongue). Table 3.2 shows basic information for compositions attributed to Bach, Beethoven, Buxtehude and Brahms, a total of 1124 music time series.

Composer (c_j)	Country of Birth	$\|R^{c_j}\|$	$E[\alpha]$	$E[H]$
Bach, Johann Sebastian	Germany	908	0.79	0.92
Buxtehude, Dieterich	Germany	21	0.88	0.94
Beethoven, Ludwig van	Germany	186	1.03	0.9
Brahms, Johannes	Germany	9	0.99	1.02

Table 3.2: Fractal Exponents - Composers born in the same country. Temporal power law exponents listed correspond to mean values. Temporal power law exponents listed here are mean values of exponents computed using the DFA (Algorithm 2.2) and Hurst (Algorithm 2.1) algorithms. $|R^{c_j}|$ denotes the number of the compositions analyzed for composer c_j.

In my investigation, I found disinctions between the fractal signatures for composers with a common mother tongue (e.g., consider Bach and Beethoven in Table 3.6).

3.4 Western Composers Case Studies

In this section, I provide a more in-depth analysis for four composers. These composers were selected for their distinct professional timelines, music styles, and the large number of music time series available for each in my Western dataset (Section 3.1). My goal is to show that the power law exponents, as measures of self-similarity, can be used to analyze and ultimately classify compositions by their composers. In other words, these power law exponents will be shown to be useful in forming a signature for each composer. The composers analyzed are: Gershwin (Section 3.4.1), Grieg (Section 3.4.2), Mozart (Section 3.4.3), and Scarlatti (Section 3.4.4). The factors influencing the creation of each composition are diverse. For instance, each composer's set of compositions is influenced by the era in which the composer lived in and his geographical location. This influence may be minimal, but I note that there is further diversity in the nationality of the composers chosen as representatives in the following sections.

Composer Name	Life Span	Professional Span	Nationality	Era
George Gershwin	1898 – 1937	1916 – 1937	American	Musical Theatre Composer
Edvard Grieg	1843 – 1907	1862 – 1906	Norwegian	Romantic
Wolfgang Amadeus Mozart	1756 – 1791	1764 – 1791	Austrian	Classical
Domenico Scarlatti	1685 – 1757	1700 – 1757	Italian	Galant

Table 3.3: Basic information about the four case study composers in my Western composer collection.

The composers selected as case studies (Table 3.3) had no direct influence on each other, lived in different time periods, hailed were from distinct countries and had various musical styles.

3.4.1 George Gershwin

George Gershwin ($1898 - 1937$) was an American composer of popular songs and musicals (Broadway shows) as well as a composer of jazz-influenced classical music [BGP06, Sch73, CSC13]. The analysis here focuses on Gershwin's professional work from 1921 to 1946. The twenty nine compositions analyzed, along with their corresponding year of composition [Sch73, Gib95, Car00], are listed in Figure 3.8. Gershwin, who by the early 1920s had become established as composer, composed his first composition in 1916 and first full Broadway score, "La, La, Lucille", in 1919 [BGP06, Sch73, CSC13]. To better understand his compositional characteristics, here I analyzed compositions from 1921 until his death.

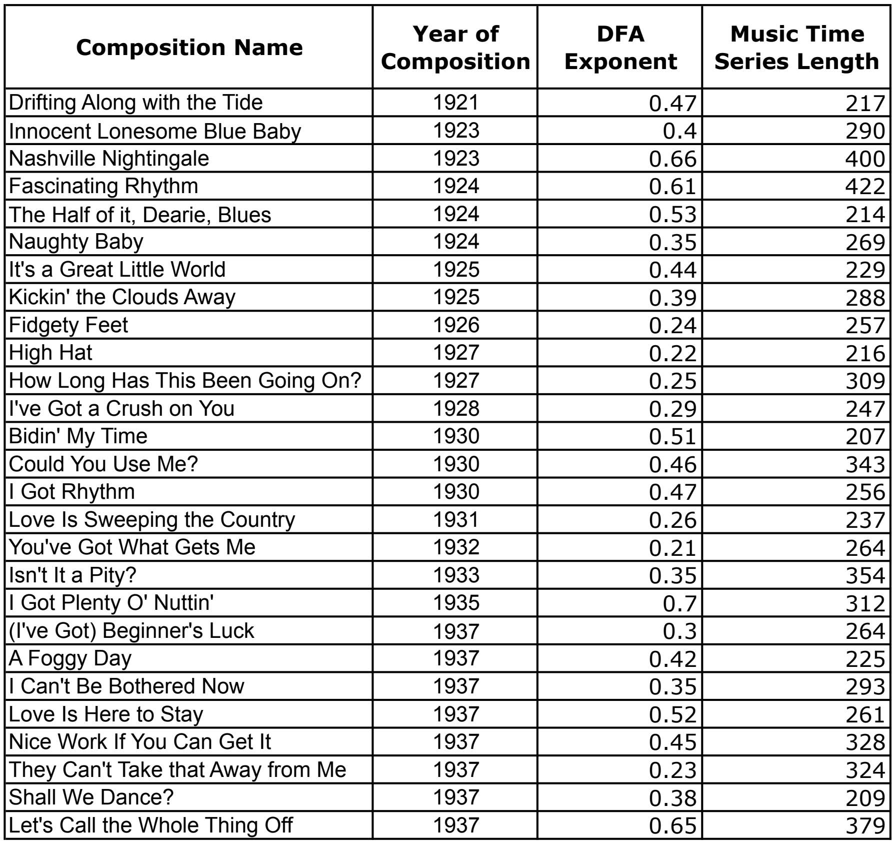
The collection available for Gershwin, $R^{\texttt{Gerwhsin}}$ contains 27 music pieces. Figure 3.8 shows the lengths of series analyzed in my investigation, when sorted by the year of composition. The two longest analyzed composition are "Fascinating Rhythm" (1924), $|r_{\texttt{gersh06.krn}}| = 422$, and "Nashville Nightingale" (1923) with $|r_{\texttt{gersh20.krn}}| = 400$. The shortest composition analyzed, $|r_{\texttt{gersh03.krn}}| = 204$, is "Bidin' My Time" composed in 1930. It also shows that, if one were to consider number of compositions in a year as a measure of activity, then — based on this small collection available — Gershwin, with four compositions, was most *active* in the year of his death.

The Gershwin music pieces, $r_i^{\texttt{Gershwin}}$, analyzed here all have DFA exponents less than

0.75: That is, Figures 3.5, 3.8 show that $\forall r_i^{\texttt{Gershwin}} \in R_{\texttt{Gershwin}}, \alpha_{r_i}^{\texttt{Gershwin}} < 0.75$. The Gershwin music pieces available — $R_{\texttt{Gershwin}}$ outlined in Figure 3.8 — were found to have much smaller DFA exponents compared to other composers included in this study (Figure 3.5); these compositions, with $\mu(\{\alpha_{r_i}^{\texttt{Gershwin}}\}_{r_i \in R_{\texttt{Gershwin}}}) = 0.41$ (Table 3.4), demonstrated very distinct temporal signatures from other composers in my analysis (Figure 3.6).

$\|R_{\texttt{Gershwin}}\| = 29$	Mean	Median	Standard Deviation
α	0.41	0.39	0.14

Table 3.4: Gershwin - Average Temporal Correlation Power Exponent (DFA)



Composition Name	Year of Composition	DFA Exponent	Music Time Series Length
Drifting Along with the Tide	1921	0.47	217
Innocent Lonesome Blue Baby	1923	0.4	290
Nashville Nightingale	1923	0.66	400
Fascinating Rhythm	1924	0.61	422
The Half of it, Dearie, Blues	1924	0.53	214
Naughty Baby	1924	0.35	269
It's a Great Little World	1925	0.44	229
Kickin' the Clouds Away	1925	0.39	288
Fidgety Feet	1926	0.24	257
High Hat	1927	0.22	216
How Long Has This Been Going On?	1927	0.25	309
I've Got a Crush on You	1928	0.29	247
Bidin' My Time	1930	0.51	207
Could You Use Me?	1930	0.46	343
I Got Rhythm	1930	0.47	256
Love Is Sweeping the Country	1931	0.26	237
You've Got What Gets Me	1932	0.21	264
Isn't It a Pity?	1933	0.35	354
I Got Plenty O' Nuttin'	1935	0.7	312
(I've Got) Beginner's Luck	1937	0.3	264
A Foggy Day	1937	0.42	225
I Can't Be Bothered Now	1937	0.35	293
Love Is Here to Stay	1937	0.52	261
Nice Work If You Can Get It	1937	0.45	328
They Can't Take that Away from Me	1937	0.23	324
Shall We Dance?	1937	0.38	209
Let's Call the Whole Thing Off	1937	0.65	379

Figure 3.8: George Gershwin $(1898 - 1937)$ - Compositions sorted chronologically by the year of composition [Sch73, Gib95, Car00].

α	Length of Analysis	Name	Year
	$(\lvert r_i^{\texttt{Gershwin}} \rvert)$		of Composition
0.66	400	"Nashville Nightingale"	1923
0.24	257	"Fidgety Feet"	1924
0.61	422	"Fascinating Rhythm"	1924
0.22	216	"High Hat"	1927
0.25	309	"How Long Has This Been Going On?"	1927
0.26	237	"Love Is Sweeping the Country"	1931
0.21	264	"You've Got What Gets Me"	1932
0.70	312	"I Got Plenty O' Nuttin"'	1935
0.23	324	"They Can't Take that Away from Me"	1937
0.65	379	"Let's Call the Whole Thing Off"	1937

Table 3.5: Gershwin - Compositions with fractal exponents at least one standard deviation removed from the mean.

The collection of Gershwin music time series studied here revealed variation in values of α, even for some compositions in the same year (e.g.,"Hight Hat" and "How Long Has This Been Going On") composed in 1927 (Table 3.5). Figure 3.9 presents the DFA exponents computed for the Gershwin sub-collection, $R_{\texttt{Gershwin}}$, sorted chronologically.

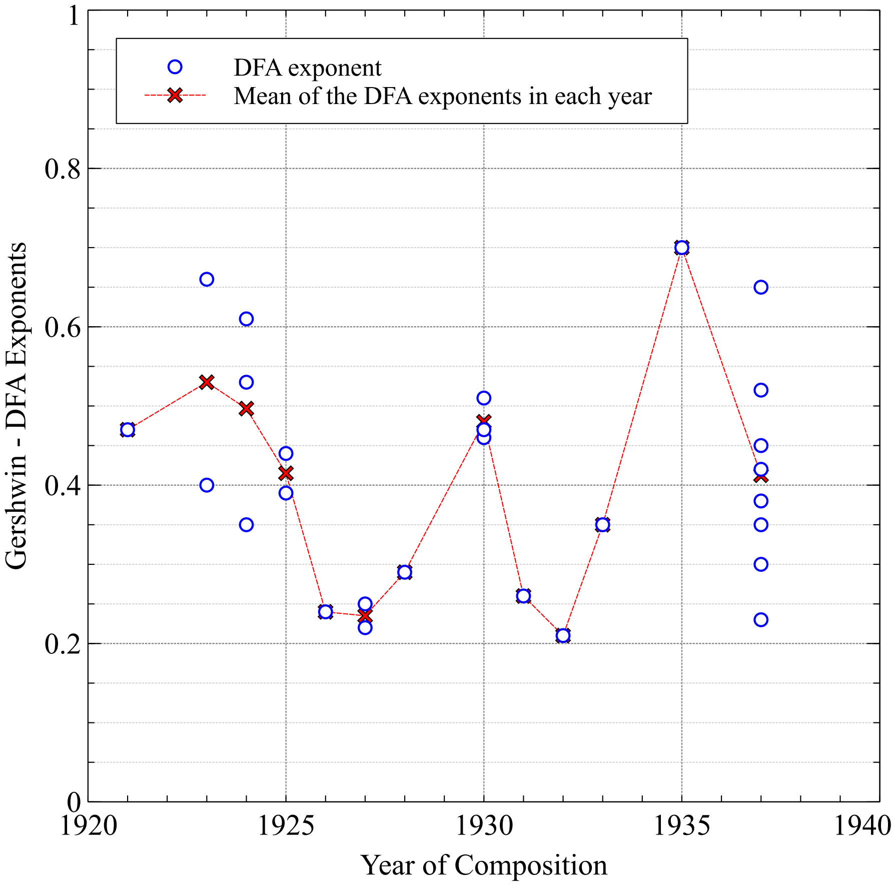


Figure 3.9: DFA exponents for the music series attributed to Gershwin in this study. The red plot illustrates the changes in mean DFA exponents — averaged over α value(s) computed for each particular year — over time.

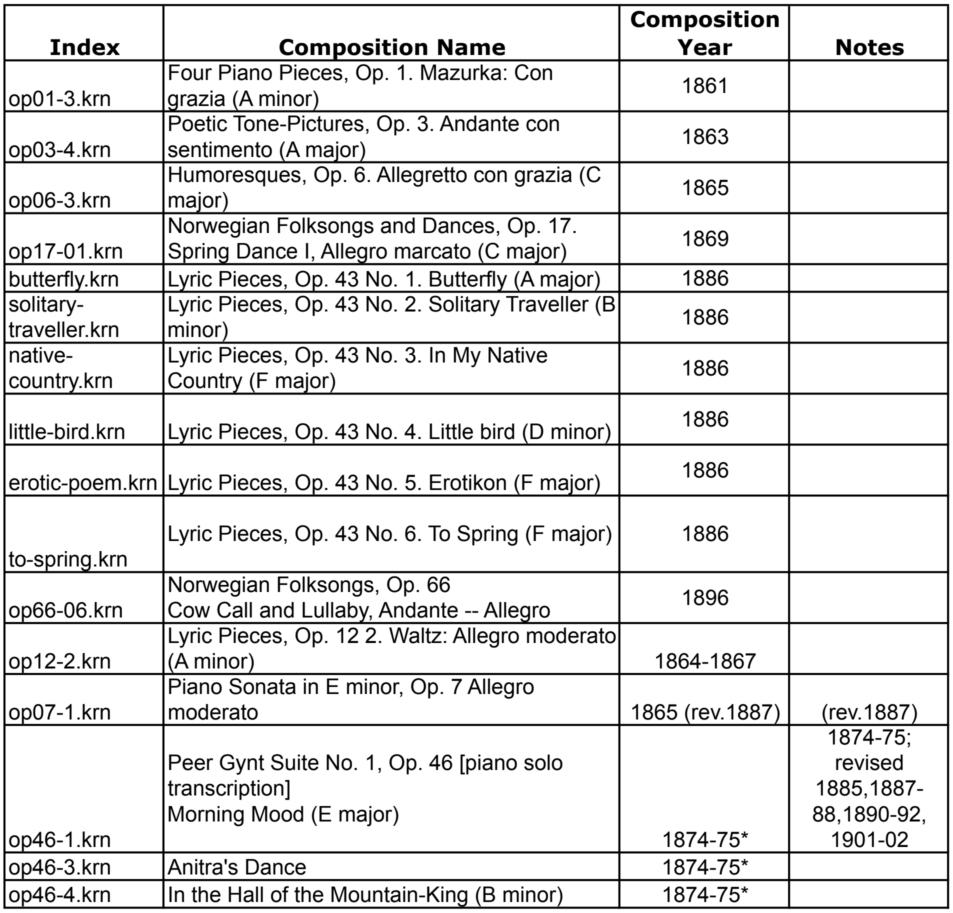
Although no clear long-range disorder trends — a consistent decrease or increase in disorder — are apparent in a chronological grouping of Gershwin's compositions (Figures 3.8 and 3.9), a few critical points of *disruption* can be detected. For instance, based on the Gershwin music time series included in this study, there is a gradual decrease in The mean α values of his compositions from $1923 - 1926$. That is, in my study, the mean

α values for compositions in 1923 to 1926 decrease; more precisely, a decreasing trend is visible from $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1923}}$ to $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1926}}$ (Figure 3.9), where the mean DFA exponents of all compositions in year y_i is computed as $\bar{\alpha}^{\texttt{Gershwin}}_{y_i} = \mu(\{\alpha^{\texttt{Gershwin}}_{r_i}\}_{r_i \in R^{\texttt{Gershwin}}_{y_i}})$. In the context of this study, Figure 3.9 also shows an increase from $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1926}}$ to $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1930}}$, a decrease from $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1930}}$ to $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1932}}$, and a increase from $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1930}}$ to $\bar{\alpha}^{\texttt{Gershwin}}_{y_{1935}}$. That is, I consider $1926, 1930,$ and 1935 to be points of disruption for this Gershwin collection (Figure 3.9). Based on these observations and in the context of the Gershwin collection available, compositions of $1926, 1930,$ and 1935 signify a sudden change in the composer's regularity style. Biographical notes indicate that Gershwin's success in 1924 led to a few lifestyle changes. The resulting financial flexibility led him to move his family to upper West side in Manhattan, and around the same time, his interest in visual arts increased as he began painting himself and also collecting paintings and sculptures [CSC13]. Years $1925 - 26$ mark the creation of his Concerto in F for piano and orchestra, which deviates from his compositional style in that it is a non-musical theater piece [CSC13]. The years following 1926 mark his growth as a composer. In 1928, he traveled to Europe and met influential composers such as Prokofiev, Ravel, Poulenc, Milhaud, and Berg [CSC13]. In particular, meeting Ravel had great influence on his compositional style [BGP06, Sch73, CSC13]. The sudden increase in both the number of compositions as well as the notable fluctuation in fractal exponents of compositions in the $1928 - 1931$ period may be attributed to a few milestones in his professional life. The early thirties mark Gershwin's focus on concert music [CSC13] in addition to his compositions for Broadway productions. The same period marks his debut as a conductor (1929) [CSC13] and his first visit to Hollywood $(1930 - 31)$ [CSC13]. This period culminated with the success of his "Porgy and Bess"

(1935) and his untimely death [CSC13]. To the extent that this Western collection permits, Gershwin was prolific both in performance and composition (Figure 3.8) in his final year [CSC13].

3.4.2 Edvard Grieg

In this section, I focus on compositions by Edvard Grieg, a Norwegian composer ($1843-1907$), included in the Western collection of Section 3.1. Compositions studied here encompass pieces from 1861 to 1875, and include solo piano, lyric piano, sonata, dances and incidental music (suite) pieces (Figure 3.10).



Index	Composition Name	Composition Year	Notes
op01-3.krn	Four Piano Pieces, Op. 1. Mazurka: Con grazia (A minor)	1861	
op03-4.krn	Poetic Tone-Pictures, Op. 3. Andante con sentimento (A major)	1863	
op06-3.krn	Humoresques, Op. 6. Allegretto con grazia (C major)	1865	
op17-01.krn	Norwegian Folksongs and Dances, Op. 17. Spring Dance I, Allegro marcato (C major)	1869	
butterfly.krn	Lyric Pieces, Op. 43 No. 1. Butterfly (A major)	1886	
solitary-traveller.krn	Lyric Pieces, Op. 43 No. 2. Solitary Traveller (B minor)	1886	
native-country.krn	Lyric Pieces, Op. 43 No. 3. In My Native Country (F major)	1886	
little-bird.krn	Lyric Pieces, Op. 43 No. 4. Little bird (D minor)	1886	
erotic-poem.krn	Lyric Pieces, Op. 43 No. 5. Erotikon (F major)	1886	
to-spring.krn	Lyric Pieces, Op. 43 No. 6. To Spring (F major)	1886	
op66-06.krn	Norwegian Folksongs, Op. 66 Cow Call and Lullaby, Andante -- Allegro	1896	
op12-2.krn	Lyric Pieces, Op. 12 2. Waltz: Allegro moderato (A minor)	1864-1867	
op07-1.krn	Piano Sonata in E minor, Op. 7 Allegro moderato	1865 (rev.1887)	(rev.1887)
op46-1.krn	Peer Gynt Suite No. 1, Op. 46 [piano solo transcription] Morning Mood (E major)	1874-75*	1874-75; revised 1885,1887-88,1890-92, 1901-02
op46-3.krn	Anitra's Dance	1874-75*	
op46-4.krn	In the Hall of the Mountain-King (B minor)	1874-75*	

Figure 3.10: List of compositions by Grieg available for my Western power law analysis. sCompositions are sorted chronologically.

First, the compositions have a mean length of 506 ± 328. The shortest composition, “Lyric Pieces, Op. 43 No. 2. Solitary Traveller (B minor)” (1886 [BSEHS88]), has 211 inter-onset durations. From his later years, I consider five compositions from 1886.

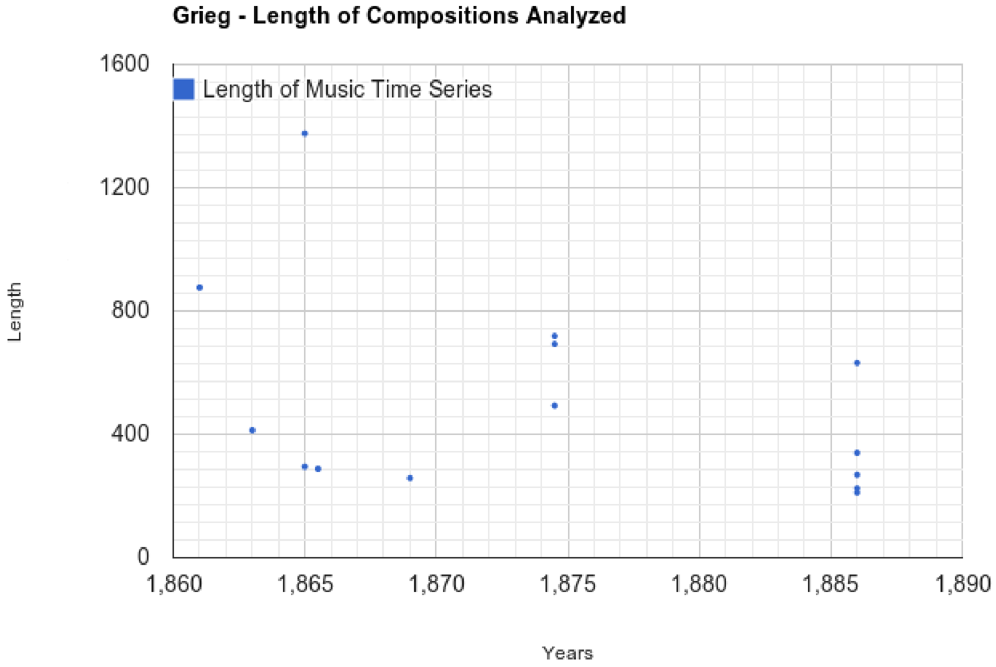


Figure 3.11: Grieg - Lengths of various music time series corresponding to compositions of Figure 3.10.

In these compositions (Figure 3.10), Grieg prefers shorter length notes. That is, the most frequent note duration used in his compositions studied here has mean value of 0.39 ± 0.22, with a range of $0.12 - 1$.

3.4.2.1 Grieg - DFA Exponent (α)

Table 3.6: Compositions with fractal exponents at least one standard deviation removed from the mean. The table also shows the length and year of composition of each music time series.

α	Index ($r_i^{\mathtt{Grieg}}$)	$\lvert r_i^{\mathtt{Grieg}}\rvert$	Composition Name	Year
1.12	op07-1.krn	1376	"Piano Sonata in E minor, Op. 7 Allegro moderato"	1865
0.41	op17-01.krn	258	"Norwegian Folksongs and Dances, Op. 17. Spring Dance I, Allegro marcato (C major)"	1869
1.07	butterfly.krn	632	"Lyric Pieces, Op. 43 No. 1. Butterfly (A major)"	1886
0.45	little-bird.krn	340	"Lyric Pieces, Op. 43 No. 4. Little bird (D minor)"	1886
0.38	solitary-traveller.krn	211	"Lyric Pieces, Op. 43 No. 2. Solitary Traveller (B minor)"	1886

In the collection analyzed here, certain pieces stand out either for their compositional context or their style. For instance, Grieg's "Norwegian Folksongs and Dances, Op. 17. Spring Dance I, Allegro marcato (C major) " (op$17-01$), composed in 1869, highlights his nationalism by incorporating Norwegian musical and composition elements [BSEHS88]. Here, his only piano sonata — dedicated to the Danish composer Niels Gade [Bai93] — is included. The piece, "Piano Sonata in E minor, Op. 7 Allegro moderato" (op07 – 1), was first composed in 1865 and then revised in 1887. The α values for both these pieces stand out from the rest and are at least one standard deviation away from the mean (Table 3.6). In particular, his piano sonata is his most structurally predictable piece with the maximal

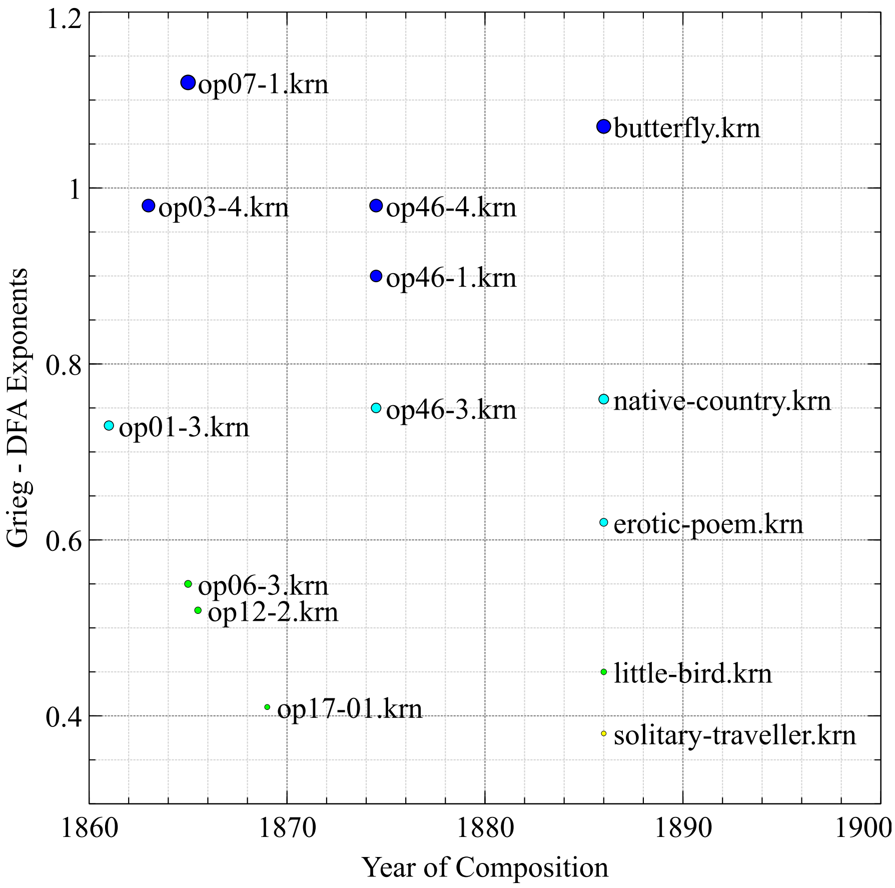


Figure 3.12: Grieg - DFA (α) exponents sorted in chronological order of composition.

α of 1.12 (Figure 3.12).

DFA analysis of Grieg's compositions (Figure 3.12) shows that the maximal α value, 1.12, occurs earlier in his professional life ("Piano Sonata in E minor, Op. 7 Allegro moderato" - 1865) when he was only 22 years old [BSEHS88]. The piece with the highest level of disorder, as measured by distance to $\alpha(\texttt{white noise}) = 0$, is "Lyric Pieces, Op. 43 No. 2. Solitary Traveller (B minor)" with an α value of 0.38 and was composed in 1886. This study contained more pieces from the composer's later years (Figure 3.12 (a)).

Here, the composer has the highest frequency of composition in 1886. All compositions in that year are lyric piano piece [BSEHS88] with "Lyric Pieces, Op. 43 No. 1. Butterfly (A major)" having the highest level of regularity ($\alpha = 1.07$), as shown in Figure 3.12 (b). From a regularity point of view, the fluctuation structure of the piece is closer to that of "Piano Sonata in E minor, Op. 7 Allegro moderato" - 1865 ($\alpha = 1.12$). That is, his second most temporally regular piece occurs in his later years (Figure 3.12). To make any definitive conclusions about chronological trends in Grieg's compositions, the study needs more compositional data. Nonetheless, Figure 3.12 represents DFA exponents computed for his compositions sorted by year, and highlights the variability of DFA exponents in his compositions, stuided here. This also shows similarities between compositions of 1863 and $1874 - 1875$, where $\alpha = 0.98$ in both cases. This demonstrates that Grieg has composed two pieces, separated by more than a decade, that are structurally highly similar.

3.4.3 Wolfgang Amadeus Mozart

> *"It is hard to think of another composer who so perfectly marries form and passion ... Mozart combines serenity, melancholy, and tragic intensity into one great lyric improvisation. Over it all hovers the greater spirit that is Mozart's-the spirit of compassion, of universal love, even of suffering–a spirit that knows no age, that belongs to all ages."*
>
> Leonard Bernstein [Ber70]

Wolfgang Amadeus Mozart ($1756 - 1791$), an iconic child prodigy, was an Austrian composer who personifies Western classical music for most (non-musician) individuals. His professional life, though short, is prolific and influential. Mozart made his first public appearance in Salzburg when he was 5 [Ste04], started composing small pieces around the same time ($1961-1962$) and composed his first symphony in 1764 [moz].

In this analysis, 169 Mozart compositions are considered. The timeline of words considered spans compositions from 1770 to his final year. Pieces considered are diverse in genre and instrument (Table 3.7):

Table 3.7: Mozart Compositions ($1770-1791$) - Genre and Instrument.

Instrument	Genre
Voice and piano (arrangement)	Song
Voice and piano (arrangement)	Opera
Piano	Variations
Piano	Sonatina
Piano	Sonata

As in the previous case studies (Sections 3.4.1 - 3.4.2), structural repetitions in each composition's sequence of durations are used to construct a Mozart *signature*. Similar to the previous case studies, fractal exponents were computed for music time series with at least 200 duration points. These music time series, 148 in total, are referred to as the Mozart collection, $R_{\texttt{Mozart}}$. The Mozart collection contains pieces of various lengths ranging from 200 to 2314.

After considering only the compositional years of the Mozart collection in my analysis, two particular years stand out: In 1773 and 1783, Mozart completed more than 20 compositions in each . In 1773, the music time series analyzed have a maximum of 961, with a mean of 578, duration elements. In 1783, his longest composition creates a music time series of length 2030, and the rest are on average 875 long. Since these two years mark years with a high number of compositions and the length of the compositions are not small, I can consider Mozart to have been abnormally active in these years.

3.4.3.1 Mozart - DFA Exponent (α)

Since it is difficult to visualize all αvalues on the same plot, Figure 3.13 illustrates mean(α) over the course of a particular year, and highlights the variability in the exponent within a particular year , and over the course of a few decades analyzed in my investigation. The overall variation in α fall within 0.89 ± 0.14 (Table 3.8). Since the compositions selected in this study span two decades and the number of compositions considered is relatively large, the small deviation from the mean for the majority of the compositions, highlights a particular style or preference. In other words, such persistence in Mozart's α values may be interpreted as a consistency in temporal structure of his compositions.

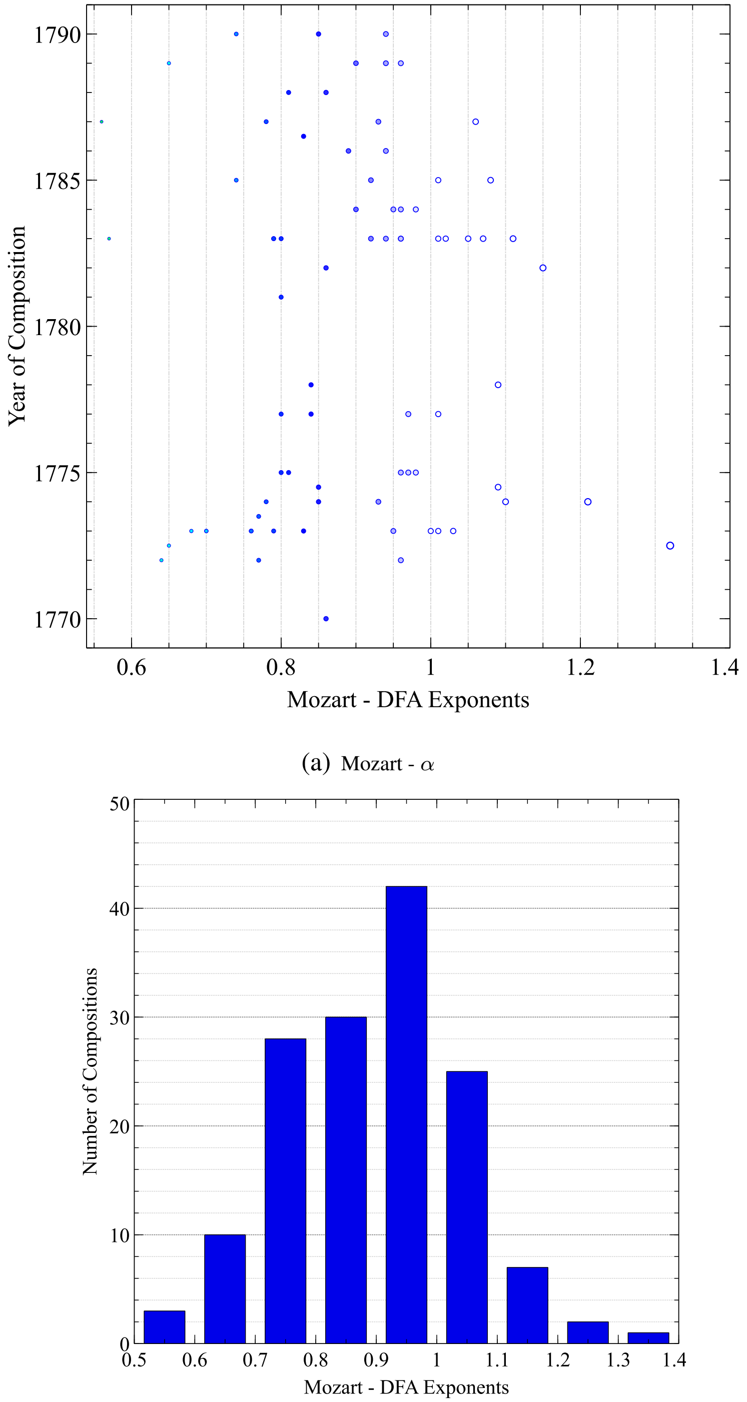
$\lvert R_{\texttt{Mozart}} \rvert = 168$	Mean	Median	Standard Deviation
α	0.89	0.90	0.14

Table 3.8: Basic Statistics computed for the DFA exponents of the Mozart music time series available in my Western collection, $\{\alpha_{r_i}^{Mozart}\}_{r_i \in R^{\texttt{Mozart}}}$.

In the collection of compositions analyzed here, Mozart's α values are closer to fractal noise (i.e., $\alpha \approx 1$) than those of Gershwin (Section 3.4.1) and Grieg (Section 3.4.2). This indicates higher regularity and structure in Mozart's compositions , which were included in this analysis. His most regular pieces in my analysis, $\alpha = \alpha_{\texttt{max}} = 1.21$, is "String Quartet No. 4 in C major" (k157-02.krn) composed in $1772 - 1773$. The other critical value, $\alpha_{\texttt{min}} = 0.54$, corresponds to his "Piano Sonata No. 1 in C major" (sonata01-2.krn) composed in $1774 - 1775$.

Despite the variations shown in Figure 3.13, various pieces have similar or approximately-similar exponents. In the collection analyzed here, Mozart's most predictable piece, $\alpha = 1.32$, is one composed in his youth (Figure 3.13), and the higher values of α, correspond to his earlier compositions. Although his compositions seem to be structurally similar — such as 40 compositions with $0.9 \leq \alpha \leq 1$ throughout his career — his more structurally predictable pieces, $\alpha > 1$, are more rare and occur in the first half of his professional life.

This analysis shows that many Mozart compositions , analyzed here, are structurally



(a) Mozart - α

(b) Number of compositions $\alpha \in (\alpha_{\mathtt{min}}^{\mathtt{Mozart}} : 0.1 : \alpha_{\mathtt{max}}^{\mathtt{Mozart}})$

Figure 3.13: Mozart - DFA (α) exponent (select compositions from $1770 - 1791$) sorted chronologically.

very similar; the number of unique fractal exponents is smaller than the number of music time series analyzed. Moreover, a large number of his compositions have highly predictable structures: $0.9 < \alpha < 1$ for more than 40 music time series (Figure 3.13). Lastly, particular structural similarities emerge repeatedly but with chronological gaps. For instance, compositions from $1775, 1777, 1781$ and 1783 all have $\alpha = 0.8$ (Figure 3.13).

3.4.4 Domenico Scarlatti

The compositions of Italian composer Domenico Scarlatti $(1685 - 1757)$ fall under the *Gallant* style: "freer, and more song-like, homophonic" than Baroque [BGP06] — and his work was influenced by Corelli, Gasparini, A. Scarlatti, and Folk music (Italian, Spanish and Portuguese) [PBH$^+$]. The Scarlatti collection includes fifty-nine of his Sonatas composed from 1738 to 1757. Henceforth, this set of compositions is referred to as the Scarlatti collection, $R^{\texttt{Scarlatti}}$.

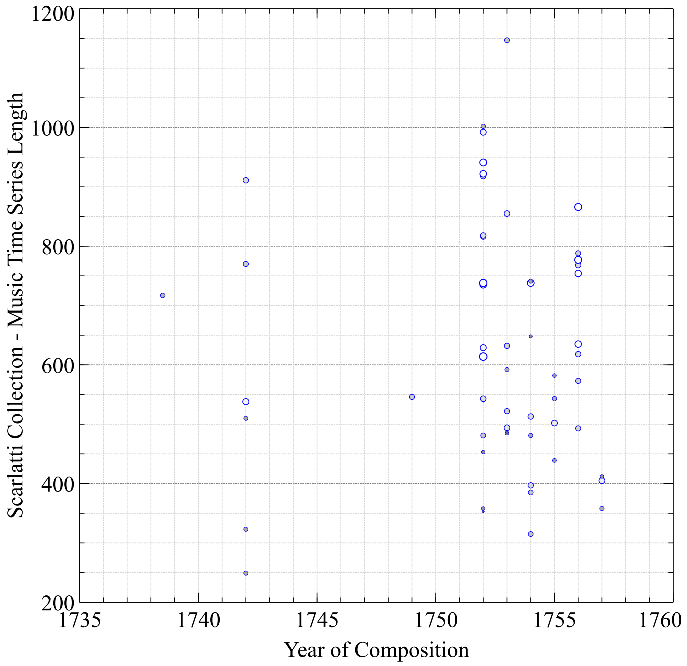


Figure 3.14: Scarlatti - Lengths of music time series generated for his compositions sorted by year of composition. The markers are scaled by the composition's corresponding α value.

Compositions analyzed here, $r_i^{\texttt{Scarlatt}} \in R_{\texttt{Scarlatti}}$, contain 639 duration points on average, $\sigma(|r_i|) = 231, \texttt{min}(|r_i|) = 249$, and $\texttt{max}(|r_i|) = 1511$.

3.4.4.1 Scarlatti - DFA Exponent (α)

Table 3.4.4.1 summarizes DFA exponents computed for the Scarlatti collection, $R_{\texttt{Scarlatti}}$.

$\|R_{\texttt{Scarlatti}}\| = 59$	Mean	Median	Standard Deviation
α	0.81	0.82	0.18

Table 3.9: Scarlatti - Temporal Correlation Power Exponent (DFA)

Figure 3.15 shows that different compositions have various scaling fluctuation exponents. That is, although the majority of compositions considered are structurally similar, some are atypical since their scaling fluctuation exponent, α is far removed from the mean (Table 3.10).

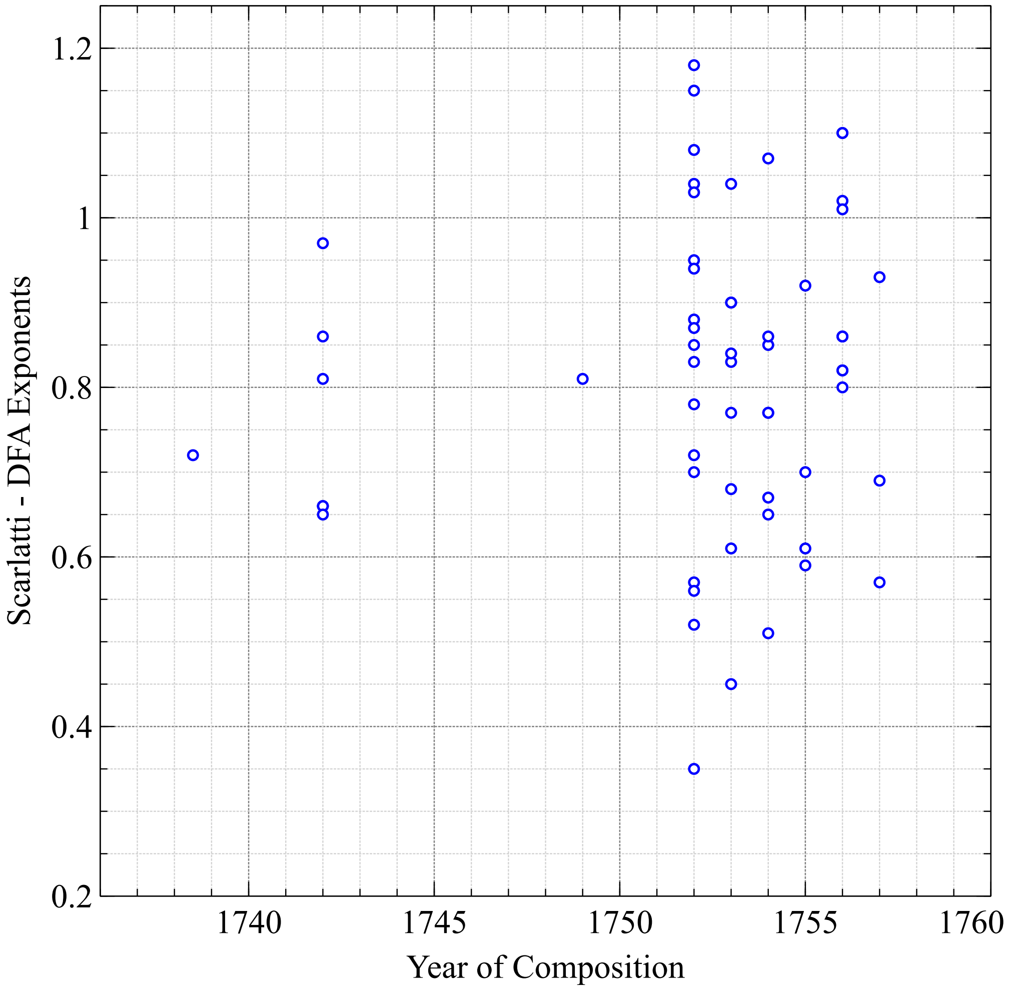


Figure 3.15: Scarlatti's Compositions $(1738 - 1757)$ - Values of α sorted chronologically.

For instance, compositions with the extremal α values are:

- $\alpha_{\mathtt{min}} = \alpha_{\mathtt{L027K238}} = 0.33$:

  "K. 238, Sonata in F minor, 4/4, Andante" (1752),

- $\alpha_{\mathtt{max}} = \alpha_{\mathtt{L348K244}} = 1.18$:

"K. 244, Sonata in B major, 3/8, Allegro" (1752).

The compositions with the most and least structural regularity were composed in the same year, and towards the end of Scarlattis life. Table 3.10 highlights compositions with structural features that are atypical in the Scarlatti collection.

	Index	Composition Name	Year of Composition	α	H
$\alpha > \mu \pm 2\sigma(R_{\texttt{Scarlatti}})$	`L198K296`	K. 296, Sonata in F major, 3/4, Andante	1753	0.77	0.86
$\alpha > \mu \pm 3\sigma(R_{\texttt{Scarlatti}})$	`L523K205`	K. 205, Sonata in F major, 2/2, Vivo	1752	0.72	0.77

Table 3.10: Scarlatti - Compositions with atypical α values

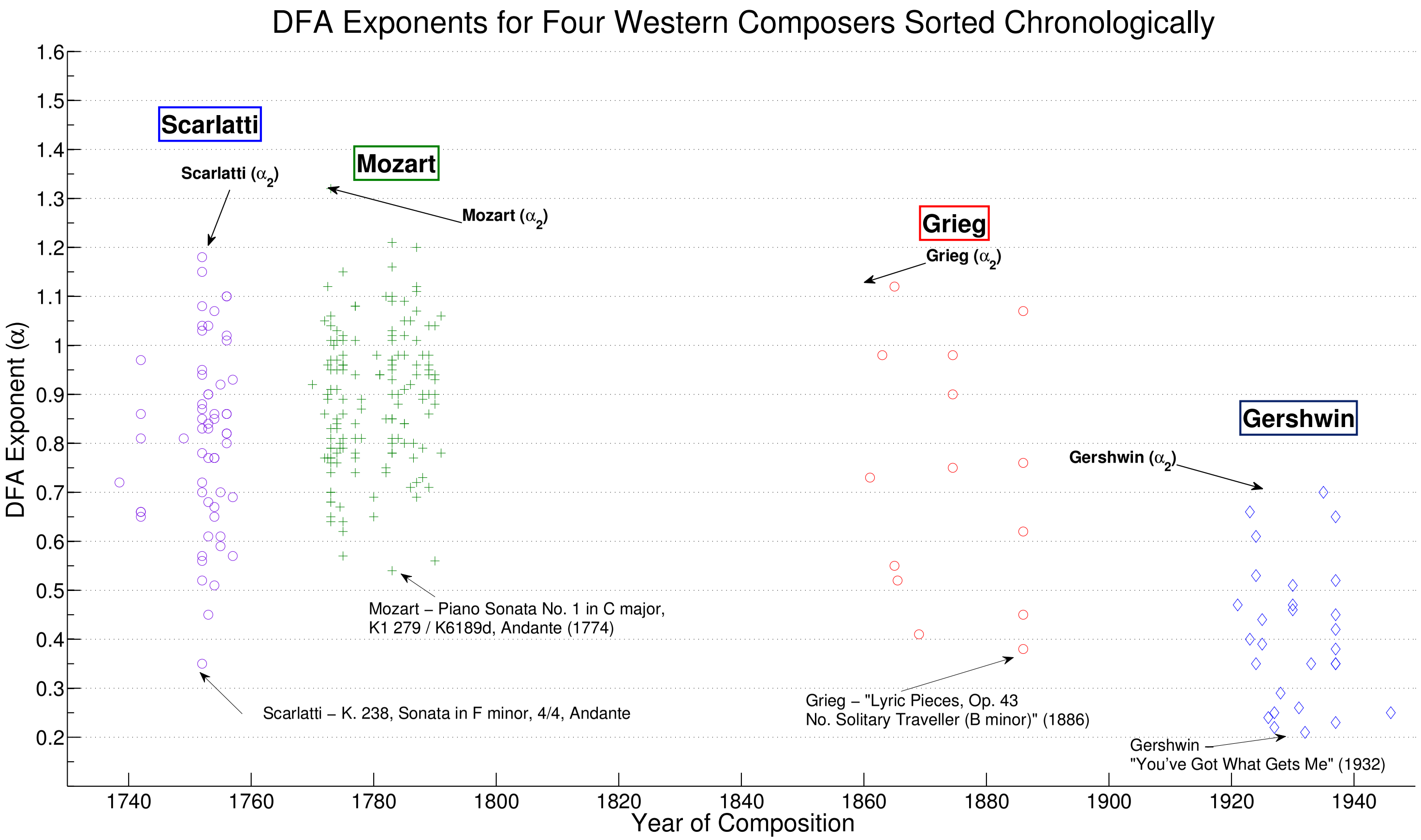


Figure 3.16: Chronologically sorted α values for compositions by Gershwin, Grieg, Mozart and Scarlatti. For each composer, compositions with minimal α values are labeled. The maximal DFA exponents marked correspond to: "I Got Plenty O' Nuttin' " (Gershwin (α_2)), "Piano Sonata in E minor, Op. 7 Allegro moderato" (Grieg (α_2)), "String Quartet No. 4 in C major, K 157 (1772-3) Andante" (Mozart (α_2)), and "K. 244, Sonata in B major, 3/8, Allegro" (Scarlatti (α_2))

Figure 3.16 presents a chronological overview for Gershwin, Grieg, Mozart and Scarlatti. It highlights that the Grieg sub-category of my music collection, the maximal $\alpha_{\texttt{Grieg}} = 1.12$ occurs earlier in his professional life ("Piano Sonata in E minor, Op. 7 Allegro moderato" 1865) [BSEHS88]; while his least predictable composition, "Lyric Pieces, Op. 43 No. 2. Solitary Traveller (B minor)" with an α value of 0.38, and was composed in 1886. From a regularity point of view, the fluctuation rhythmic structure of "Lyric Pieces, Op. 43 No. 1. Butterfly (A major)" is very similar to "Piano Sonata in E minor, Op. 7 Allegro moderato" 1865 ($\alpha = 1.12$). The 168 Mozart compositions selected in this study span over two decades, and the concentration of α values in Figure 3.16 over the course of the $1770 - 1790$ highlights a consistency in his compositional preferences; for instance, there are 40 Mozart pieces with $0.9 \leq \alpha \leq 1$, and that the majority of his compositions fell in the $0.5 - 0.7$ range (Figure 3.17). Over time, there is a slight decrease in mean α-values computed for the rhythm of compositions by Mozart. His most regular piece in this analysis — "String Quartet No. 4 in C major" has $\alpha = 1.21$ — was composed in $1772 - 1773$; whereas, the least structured composition — in $1774 - 1775$. Finally, the rhythmic structures in Scarlatti's compositions, sorted chronologically in Figure 3.16, are clustered in the $0.75 - 0.85$ range for the majority of compositions considered here In this study, the compositions with the most and least structural regularity were composed in the same year, and towards the end of Scarlatti's life: $\alpha_{\texttt{min}} = 0.33$ ("K. 238, Sonata in F minor, 4/4, Andante" 1752), and $\alpha_{\texttt{max}} = 1.18$ ("K. 244, Sonata in B major, 3/8, Allegro" 1752).

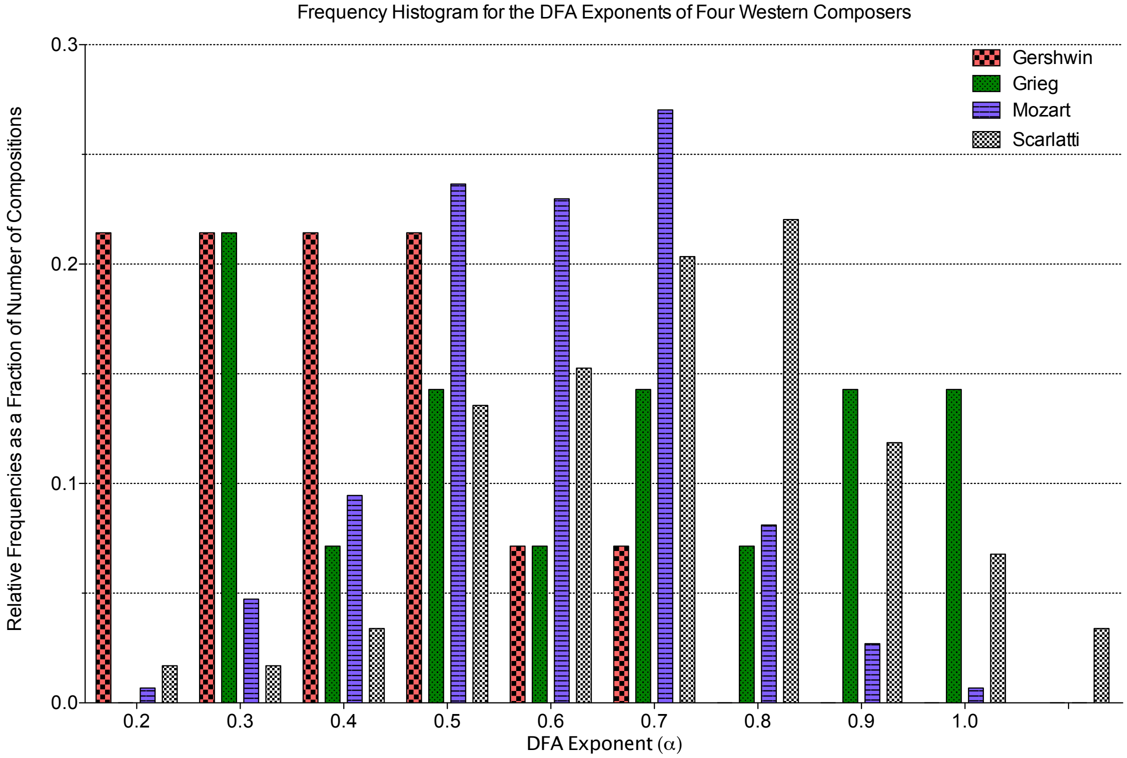


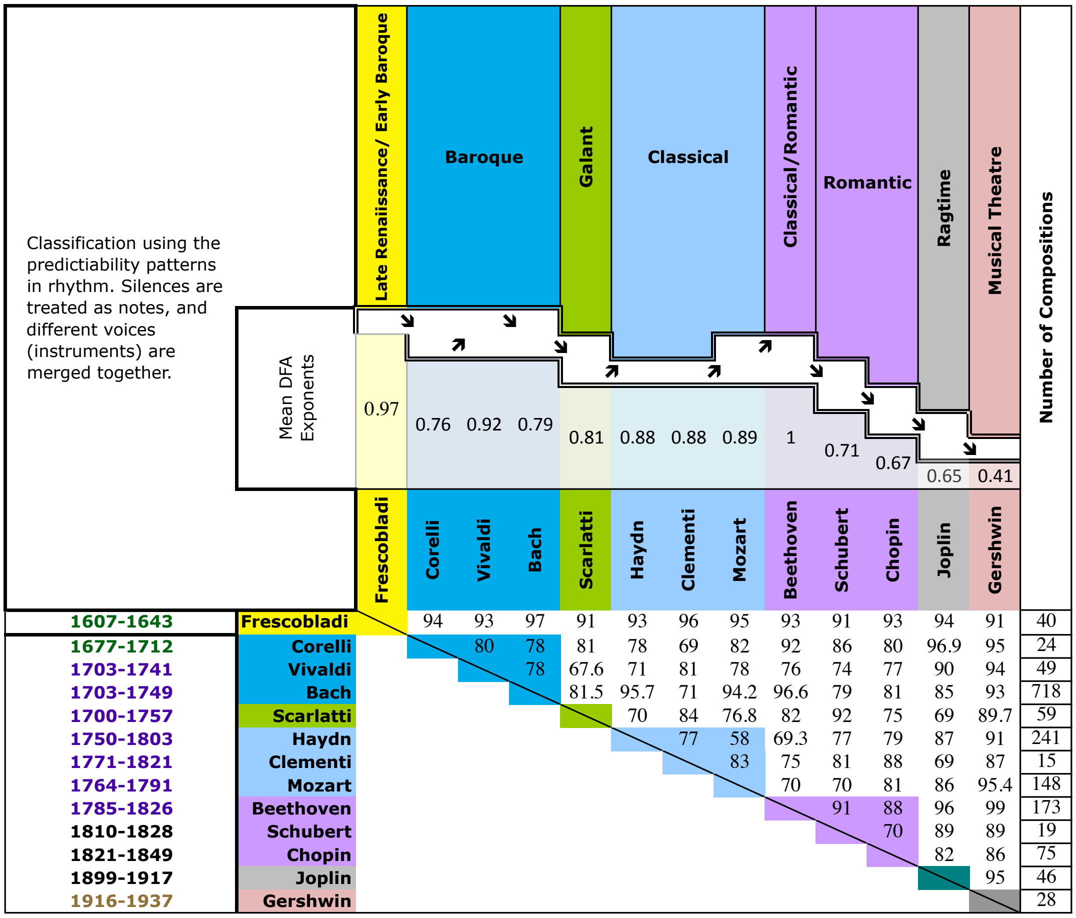
Figure 3.17: Each bar represents the normalized number of compositions with a particular value of DFA exponent (α), for Gershwin, Grieg, Mozart and Scarlatti.

3.5 Binary Classification

> *"All models are wrong, but some are useful."*
>
> - George E. P. Box [BD87]

This section presents classification results using temporal scale-free features of structural repetitions of Western compositions studied in this chapter. The accuracy of distinguishing between two Western composers using such temporal self-similarity power law exponents is presented as evidence of their significance as descriptors. In other words, classification accuracy is interpreted as an indicator for rich information content captured by these fractal exponents. Here, Western compositions classified are grouped by composers, styles, or period. I use binary classification as outlined in Section 2.4.4): pairs of the Western composition groupings are classified using logistic regression and decision trees. As discussed in Section 2.4.4, a particular implementation of decision tree classifiers, $J48$ [Qui93], is used in my analysis. In all classification cases, WEKA's user interface was directly used [HFH$^+$09] (Section 2.4.4).

These classifiers train on fractal exponents and predictability features listed in Table 2.4.4. Fractal exponents consist of spectral, DFA and Hurst exponents computed for each composition's music time series. For each pair of composers, the confusion matrix, presented in percentages, further highlights what portion of a composer's compositions were classified correctly. A 10-fold cross-validation is used for both classifiers, and unless otherwise specified, no other filtering has been applied. When classifying two sets which differ in size by more than a factor of 2.5, the larger set has been subsampled randomly. Mean absolute error (MAE) and weighted F-measures are statistical measures of accuracy of the classification (Section 2.4.4).



Classification using the predictiability patterns in rhythm. Silences are treated as notes, and different voices (instruments) are merged together.

		Late Renaissance/ Early Baroque	Baroque			Galant	Classical			Classical/Romantic	Romantic		Ragtime	Musical Theatre	Number of Compositions
	Mean DFA Exponents	0.97	0.76	0.92	0.79	0.81	0.88	0.88	0.89	1	0.71	0.67	0.65	0.41	
		Frescobladi	Corelli	Vivaldi	Bach	Scarlatti	Haydn	Clementi	Mozart	Beethoven	Schubert	Chopin	Joplin	Gershwin	
1607-1643	Frescobladi		94	93	97	91	93	96	95	93	91	93	94	91	40
1677-1712	Corelli			80	78	81	78	69	82	92	86	80	96.9	95	24
1703-1741	Vivaldi				78	67.6	71	81	78	76	74	77	90	94	49
1703-1749	Bach					81.5	95.7	71	94.2	96.6	79	81	85	93	718
1700-1757	Scarlatti						70	84	76.8	82	92	75	69	89.7	59
1750-1803	Haydn							77	58	69.3	77	79	87	91	241
1771-1821	Clementi								83	75	81	88	69	87	15
1764-1791	Mozart									70	70	81	86	95.4	148
1785-1826	Beethoven										91	88	96	99	173
1810-1828	Schubert											70	89	89	19
1821-1849	Chopin												82	86	75
1899-1917	Joplin													95	46
1916-1937	Gershwin														28

Figure 3.18: Classification accuracy (%) of distinguishing between distinct Western composer using decision tree algorithm J48 [HFH+09] as binary classifiers. The various musical eras are highlighted in different colours, and composers from the same century are boxed together.

The classification accuracy results for all pairs of composers (using the decision tree algorithm) are reported in Figure 3.18. The composers are sorted in increasing chronology, and for each composer, the appropriate musical style and period of professional activity are highlighted. Overall, the accuracy classifications are quite high. Figure 3.19 visually conveys that Gershwin and Clementi yielded the best and the worst mean classification accuracies, respectively.

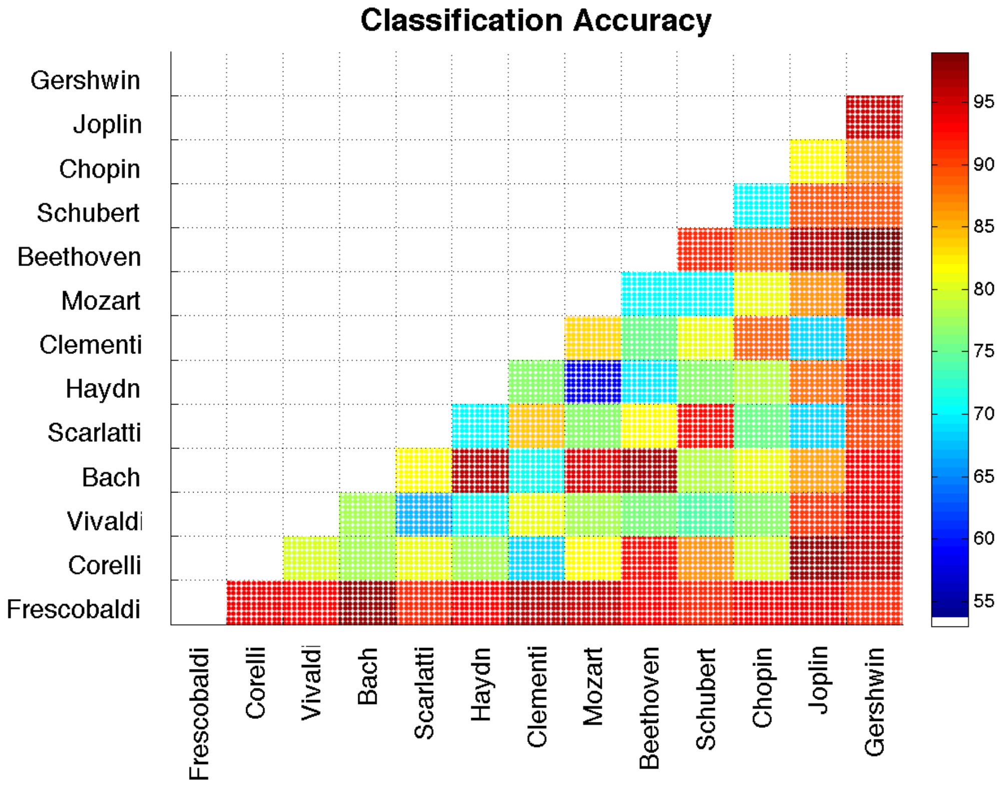


Figure 3.19: Binary classification accuracy colormap, wherein the colours represent various values of classification accuracy. The composers are sorted in increasing chronological order from Frescobaldi to Gershwin.

In the following sections, I consider various groupings of composers to better understand the influence of a composer's musical era, coeval composers and geographical origin on classification accuracy. Further details about the classification results in each category are included in Appendix B.

The precise accuracy, A_{c_i,c_j}, of correctly classifying music pieces for all possible pairs of composer labels, $c_i \in R^{\texttt{Western}}$ and $c_j \in R^{\texttt{Western}}$, are shown in Figure 3.18. Gershwin yielded the best mean classification accuracy in this study: $\bar{A}_{\texttt{Gershwin}} \geq \bar{A}_{c_i} \forall i$, where the mean classification accuracy is computed as $\bar{A}_{c_i} = \frac{1}{N} \Sigma_{c_j \neq c_i} A_{c_i,c_j}$ (Figure 3.18). To better understand the influence of a composer's era, his coeval composers and artists, and geographical origin on his corresponding signature, we analyzed the following groupings of composers.

3.5.1 Disparate Composers

The classification analyses here considered composers from $R_{\texttt{Western}}$ with different musical eras, whom hailed from distinct countries and lived in distinct time periods (e.g., Corelli vs. Frescobaldi). In the context of this study, the resulting classification is highly accurate. Figure 3.18 shows that

- $A_{\texttt{Corelli,Joplin}} = 97\%$, where Corelli $(1677 - 1712$, Baroque) - Joplin $(1899 - 1917$, Ragtime),
- $A_{\texttt{Mozart,Gershwin}} = 95\%$, where Mozart $(1764 - 1791$, Classical) - Gershwin $(1916 - 1937$, Musical Theatre),
- $A_{\texttt{Frescobaldi,Vivaldi}} = 93\%$, where Frescobaldi $(1607 - 1643$, Late Renaissance/Early Baroque) - Vivaldi $(1703 - 1741$, Baroque),

- $A_{\texttt{Scarlatti,Gershwin}} = 90\%$, where Scarlatti (1700−1757, Gallant) - Gershwin (1916−1937, Musical Theatre).

3.5.2 Stylistically Affiliated Composers

Two composers, c_i and c_j, are considered stylistically affiliated if musical history has some record of notable influence by composer c_i on c_j, or vice versa. For instance, Mozart and Haydn are stylistically affiliated composers because of Haydn's influence on Mozart's compositional style [Bro81]. Figure 3.11 lists a few stylistically affiliated composers in this study. The classification results — computed for the pairs of composer labels in this $R_{\texttt{Western}}$ — are less accurate (Figure 3.18).

	Professional Life	Musical Era	$E[\alpha]$	$E[H]$
Bach	1703 – 1749	Baroque	0.79	0.92
Haydn	1750 – 1803	Classical	0.88	0.87
Mozart	1764 – 1791	Classical	0.89	0.84
Joplin	1899 – 1917	Ragtime	0.65	0.73
Sousa	1872 – 1932	Romantic/Military	0.82	0.93

Table 3.11: DFA and Hurst exponents for interdependent composers

For instance, $A_{\texttt{Bach,Vivaldi}} = 78\%$ (Baroque) and $A_{\texttt{Clementi,Mozart}} = 58\%$ (Classical). For more details on such classifications, see Appendix B.

3.5.3 Coeval Composers

Coeval composers may have similar musical styles, sources of inspiration, historical constraints or major events (e.g., wars or other national atrocities). Table 3.12 highlights basic fractal features of three coeval composers included in this study.

Table 3.12: DFA and Hurst exponents for composers with similar timelines.

Composer	Professional Life	Life Timeline	Musical Era	$E[\alpha]$	$E[H]$
Bach	$1703 - 1749$	$1685 - 1750$	Baroque	0.79	0.92
Scarlatti	$1700 - 1757$	$1685 - 1757$	Galant	0.92	0.95
Vivaldi	$1703 - 1739$	$1678 - 1741$	Baroque	0.81	0.87

Based on the music samples of this study, it appears that highly accurate classifications are possible in this case (Figures 3.18 and 3.19). For instance, $A_{\texttt{Bach,Haydn}} = 96\%$, $A_{\texttt{Bach,Beethoven}} = 97\%$, and $A_{\texttt{Bach,Scarlatti}} = 82\%$; while in cases of high fractal similarities, distinguishing between the two composers proved more difficult (e.g., $A_{\texttt{Beethoven,Haydn}} = 69\%$, $A_{\texttt{Scarlatti,Vivaldi}} = 68\%$, and $A_{\texttt{Mozart,Scarlatti}} = 77\%$). Finally, Table 3.13 summarizes binary classification results for Bach, Scarlatti and Vivaldi using decision trees. Similar accuracy results — Bach - Vivaldi: 67%; Vivaldi - Scarlatti: 88.9% — are obtained using logistic regression.

Table 3.13: Binary Classification - Composers from similar time periods. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Bach}}\rvert = 27, \lvert R_{\texttt{Vivaldi}}\rvert = 49$				Bach	Vivaldi	
Bach $(1685 - 1750)$ - Vivaldi	77.6	0.23	0.78	19	8	Bach
				9	40	Vivaldi
$\lvert R_{\texttt{Bach}}\rvert = 22, \lvert R_{\texttt{Scarlatti}}\rvert = 59$				Bach	Scarlatti	
Bach - Scarlatti $(1678 - 1741)$	81.5	0.22	0.81	12	10	Bach
				5	54	Scarlatti
$\lvert R_{\texttt{Vivaldi}}\rvert = 49, \lvert R_{\texttt{Scarlatti}}\rvert = 59$				Vivaldi	Scarlatti	
Vivaldi $(1678 - 1741)$ - Scarlatti	67.6	0.33	0.67	28	21	Vivaldi
				14	45	Scarlatti

3.5.4 Composers with Identical First Language

In this section, Western composers hailing from the same country, in this music collection, are classified. Patel [Pat06] shows that the native language of a composer and the structure of his compositions may be correlated. Moreover, results from Section 3.3.2 demonstrated similarities between composers who share a common first language because of their country of origin (Table 3.2). Here, I present classification results for two German composers (Bach versus Beethoven) and two Italian composers (Frescobaldi versus Vivaldi).

Table 3.14: Binary Classification - Composers with Similar Country of Origin. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Bach}}\rvert = 241, \lvert R_{\texttt{Beethoven}}\rvert = 173$				Bach	Beethoven	
Bach - Beethoven	96.6	0.04	0.97	232	9	Bach
				5	168	Beethoven
$\lvert R_{\texttt{Vivaldi}}\rvert = 49, \lvert R_{\texttt{Frescolbaldi}}\rvert = 40$				Vivaldi	Frescobaldi	
Vivaldi - Frescobaldi	93.3	0.08	0.93	48	1	Vivaldi
				5	35	Frescobaldi

The high classification accuracy evident in Figures 3.18 and 3.19 — as well as further details included in Table B.1 — demonstrates that composers with similar language heritage in $R_{\texttt{Western}}$, may vary in their structural compositional style. This is in contrast to the high classification accuracy yielded for the following two pairings in this analysis: $A_{\texttt{Bach,Beethoven}} = 96.6\%$ and $A_{\texttt{Frescobaldi,Vivaldi}} = 93.3\%$.

3.5.5 Compositions from Different Centuries

To show the power of these exponents in classification, I consider three eras for binary classification. For each composer included in groups $1 - 3$, I indicate the range of years of his professional activity in brackets. I also note that the span of a composer's professional life does not reflect the same range of my analysis. For all composers, my analysis is a subset of their compositions.

1. **Early to mid seventeenth century:** Giovannelli $(1583-1624)$, Monteverdi $(1582-1643)$, Frescobaldi $(1607-1643)$.

2. **Mid to late eighteenth century:** Scarlatti $(1700-1757)$, Haydn $(1750-1803)$, Mozart $(1764-1791)$.

3. **Early to mid twentieth century:** Sousa $(1872-1932)$, MacDowell $(1880-1904)$, Scriabin $(1886-1914)$, Gershwin $(1916-1937)$.

To demonstrate the efficacy of temporal power law exponents in century identification, we grouped compositions into three groups, labeled by their corresponding century of composition. We found that accurate century identification based on fractal exponents is possible, $A_{1583-1643,1700-1757} = 90.6\%$. In such cases, the classifier was able to correctly assign century labels because the structural patterns, quantified by fractal exponents in musical rhythm, are significantly distinguishable; this is in contrast to $A_{1700-1791,1872-1937}] = 78\%$, which signifies the structural similarities between compositions in of the $17^{\mathtt{th}}$ century and those of the early 18 to the $19^{\mathtt{th}}$ centuries results, in this study.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Group1}}\rvert = 59, \lvert R_{\texttt{Group3}}\rvert = 58$				1600 – 1650	1900 – 1950	
Group 1 - Group 3	89.7	0.11	0.9	51	7	1600 – 1650
				5	54	1900 – 1950
$\lvert R_{\texttt{Group1}}\rvert = 59, \lvert R_{\texttt{Group2}}\rvert = 69$				1600 – 1650	1750 – 1800	
Group 1 - Group 2	90.6	0.11	0.91	61	8	1600 – 1650
				4	55	1750 – 1800
$\lvert R_{\texttt{Group2}}\rvert = 69, \lvert R_{\texttt{Group3}}\rvert = 58$				1750 – 1800	1900 – 1950	
Group 2 - Group 3	78	0.25	0.78	41	17	1750 – 1800
				11	58	1900 – 1950

Table 3.15: Binary Classification - Compositions from the $17^{\texttt{th}}, 18^{\texttt{th}}$ and $20^{\texttt{th}}$ centuries. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

3.5.6 Influence of Fractal Exponents on Binary Classification

In this section, binary classification using $J48$ decision trees of the following pairs of Western composers are considered in greater detail:

- Gershwin - Mozart (Table 3.16),
- Gershwin, Scarlatti (Table 3.17), and
- Mozart, Scarlatti (Table 3.18).

Table 3.16: Binary Classification: Gershwin - Mozart. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\|R_{\texttt{Gershwin}}\| = 28, \|R_{\texttt{Mozart}}\| = 59$				Gershwin	Mozart	
Fractal Exponents, Π	95.4	0.05	0.95	26 2	2 57	Gershwin Mozart
Π	94.3	0.07	0.94	26 3	2 56	Gershwin Mozart
Fractal Exponents	95.4	0.05	0.95	26 2	2 57	Gershwin Mozart
α	93.1	0.09	0.93	23 1	5 58	Gershwin Mozart

Table 3.16 shows binary classification of Gershwin versus Mozart. Mozart's feature space is randomly sub-sampled to ensure that both training sets, $R_{\texttt{Gershwin}}, R_{\texttt{Mozart}}$, are comparable in size. The classification accuracy is high, and the high accuracy classification results of using only Fractal exponents, or only α, highlight distinct differences in rhythmic structure of music composed by Gershwin and Mozart. A similar observation applies to the classification of Gershwin and Scarlatti. These demonstrate that the more unpredictable musical structure in Gerswhin compositions — $E[\alpha] = 0.41, E[H] = 0.49$ — contain sufficient information to be used by classifiers to distinguish his compositions effectively from those of others.

Table 3.17: Binary Classification: Gershwin - Scarlatti. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Gershwin}}\rvert = 28, \lvert R_{\texttt{Scarlatti}}\rvert = 59$				Gershwin	Scarlatti	
{Fractal Exponents, Π}	89.7	0.11	0.9	24	4	Gershwin
				5	54	Scarlatti
Π	89.7	0.11	0.9	22	6	Gershwin
				3	56	Scarlatti
Fractal Exponents	86.2	0.14	0.86	23	5	Gershwin
				7	52	Scarlatti
α	87.4	0.19	0.87	21	7	Gershwin
				4	55	Scarlatti

Finally, within the parameters of our music dataset in this study, using rhythmic structure of Mozart's compositions ($E[\alpha] = 0.89, E[H] = 0.84$) here with Scarlatti's ($E[\alpha] = 0.81, E[H] = 0.87$) to distinguish between the two groups is not as successful as the comparison of each with Gershwin. In this case, the lower classification success rate of $R_{\texttt{Mozart}} - R_{\texttt{Scarlatti}}$ (Table 3.18) — compared to their corresponding binary classification with other composers — highlights temporal similarities in rhythmic patterns of the $R_{\texttt{Mozart}}, R_{\texttt{Scarlatti}}$ collections.

Table 3.18: Binary Classification: Mozart - Scarlatti. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Mozart}}\rvert = 148, \lvert R_{\texttt{Scarlatti}}\rvert = 59$				Mozart	Scarlatti	
Fractal Exponents, Π	76.8	0.24	0.77	126 23	22 33	Mozart Scarlatti
Π	68.6	0.38	0.65	131 48	17 11	Mozart Scarlatti
Fractal Exponents	76.3	0.28	0.76	129 30	19 29	Mozart Scarlatti
α	70	0.39	0.65	135 49	13 10	Mozart Scarlatti

3.6 Conclusion and Future Works

The work presented in this chapter was motivated by the existence of fractal exponents in rhythm [LCM12]. I adapted methodology used in [LCM12] (Section 2.4), and analyzed fractal exponents of various compositions. I verified the results of [LCM12] by implementing Hurst [Hur51, HBS65] and Detrended Fluctuation Analysis [PHSG95, PBH$^+$94, KKBR$^+$01] exponents (α and H respectively) for each composition in a Western collection comprising of 1165 Western classical music pieces (Section 3.1). Novel contributions of this chapter include: granular analyses of fractal exponents of twenty four Western composers; these exponents were further statistically analyzed and used for composer classification. The results suggest that fractal exponents capture essential structural aspects of music.

The composer-by-composer analysis highlighted structural variation latent in temporal power law exponents of different compositions by a particular composer. Though these exponents are typically clustered around the mean fractal exponent value, there are anomalies. The degree of variation in structural regularity — as captured by fractal exponents — depends on the composer analyzed. For better understanding of this variation, I focused on four Western classical composers from different eras and with diverse compositional style. More precisely, compositions by George Gershwin $(1898 - 1937)$, Edvard Grieg $(1843 - 1907)$ from the Romantic era, Wolfgang Amadeus Mozart $(1756 - 1791)$ from the Classical era, and Domenico Scarlatti $(1685 - 1757)$ from the Galant musical style were analyzed. The four composers are geographically diverse; they hail from the USA, Norway, Austria and Italy respectively. I presented a detailed analysis of α for these composers, and highlighted compositions that had anomalous exponents (with corresponding biographical notes when available). Temporal power law exponents were used for classification. The classification accuracy results demonstrated that fractal exponents carry enough information to be used for decade, composer and genre classification. In other words, this work presented further evidence that temporal power laws in rhythm carry critical information that enable the use of such exponents in classification, identification (decade, composer or classical music genre). In the future, it may be feasible to search or recommend music based on structural similarities to an input music piece.

4

Power Law Signatures for Non-Western Compositions

“Music is the universal language of mankind.”

- Henry W. Longfellow [Lon86]

4.1 Introduction

Though music is prevalent across cultures, there is no consensus on the origins of its universality [Net92, BBN95]. Some writers have accounted for the ubiquity of music by considering it to be a byproduct of the innate biological characteristics of its creators [WMB01, Per06] while others consider music to be a culture-dependent social vehicle, influenced by a particular culture’s fundamental values [Sch75]. Leonard Bernstein, based on Montaigne’s notion of “universality in diversity” [DM77], investigated this duality by understanding the underlying similarities and candidate universal features in music [Ber76]:

> Just as the grammars of human languages (even mutually unintelligible ones) may have sprung from the same monogenetic sources, so in the same way

> highly varied musical tongues (which are also strangers to one another) can be said to have developed out of their common origins.

This chapter studies power law features in the rhythmic structure of music originating from four culturally diverse countries. The collection analyzed here is categorized by geographical origin and consists of samples from Africa, China, Iran and Turkey. For simplicity, I label and refer to this collection as the "non-Western collection", though this is not a comprehensive collection or a complete representative of rich, non-Western music.Temporal scale-free exponents are used to form *signatures* for their corresponding categories in the non-Western collection. This chapter attempts to determine whether there is a commonality between the rhythmic structure in non-Western and Western music analyzed in thesis, and to determine whether scale-free structural descriptors are suitable for automatically distinguishing between the dichotomies. That is, I use these scaling signatures in binary classifications of different categories of non-Western music; this demonstrates that such structural features impart significant information.

4.2 Methods and non-Western Music Collection

Though power laws in the musical rhythm of Western compositions have been previously studied by Levitin et al. [LCM12] and in Chapter 3 of this thesis, such an investigation for non-Western music is entirely novel. The methodology of analysis adopted here, analogous to Chapter 3, is outlined in Section 2.4.

4.2.1 Related Works

Music, though prevalent in all cultures [Bro91, BBN95], manifests itself with cultural specificity. Moreover, fundamental features of music, such as pitch and rhythm structures, vary widely across different cultures [Net56], and some features may not be universally detectable by all listeners [Jeh05]. However, even the casual listener can notice stylistic differences between Western and non-Western music in terms of scales, tonality, and rhythm. That is, even without being able to identify or label these attributes, most listeners say that these musics "feel" different. A compelling research question is whether the mathematical analysis techniques developed in the context of Western music can be applied to the music of other cultures [TKSW07]. The existence of large ethnic collections, such as traditional Indian [Cho07], African [CDCDT$^+$05], and Turkish Makam [Kar12], make such an investigation possible. For instance, rhythmic features have been used for music identification of Greek and African traditional music [APT$^+$07], style classification of Malay music [NDW05], and the analysis of Persian "Santur" music [HRQ05].

4.2.2 Non-Western Dataset

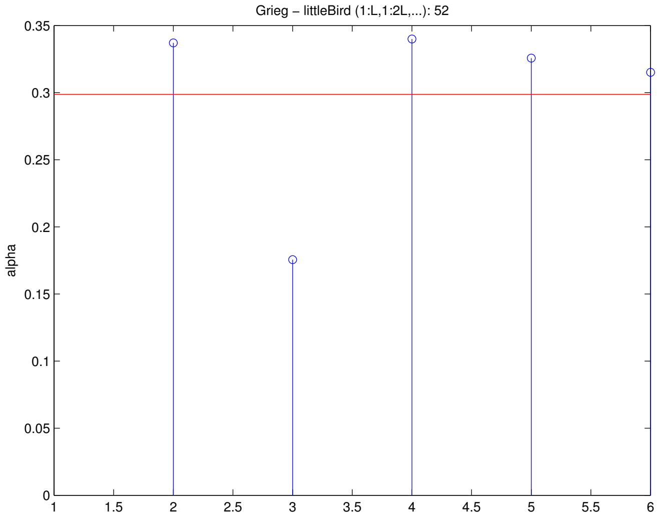
The non-Western symbolic music collection, denoted by $R_{\texttt{non-Western}}$, consists of African, Chinese, Persian and Turkish music samples, that are encoded in some appropriate machine-readable format. The African and Chinese datasets are from the KernScores collection [Sap05] (Humdrum **kern); the Persian collection (MIDI) is created by Abdoli [Abd11], and the Turkish collection (MIDI) considered is the "SymbTr Collection" [Kar12]. The four regional labels — African, Chinese, Persian and Turkish — can be used to divide and analyze the resulting sub-categories in $R_{\texttt{nonWestern}}$. The choices of the music samples in this collection are limited by availability in the corresponding source music library; the collection is not all inclusive or well-representative of all non-Western music. However, analyses presented in the following section can be used as a stepping stone for future, more comprehensive analyses of symbolic music.

Similar to the Western analysis of Chapter 3, each music piece included in $R_{\texttt{non-Western}}$ is represented as a music time series: that is, a sequence of inter-onset durations of notated musical events, notes and silences, ordered in time (Section 2.4.1). Although this collection of symbolic non-Western music has over 4400 pieces, the majority of their corresponding series are quite short with $|r_i^*| < 100$.

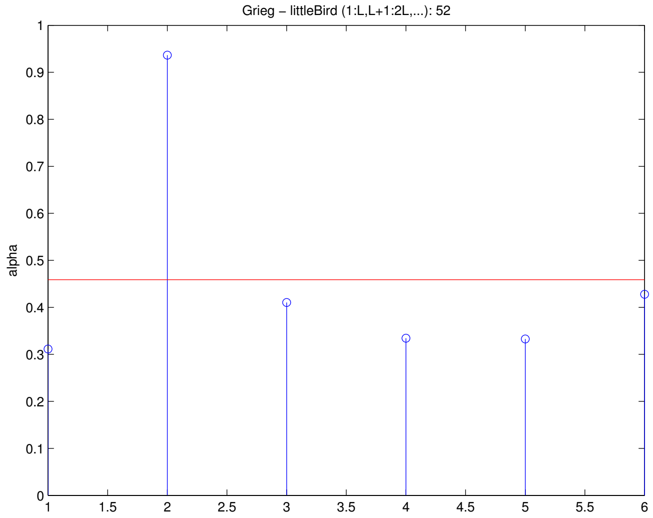
c_j	$\lvert R^{c_j}\rvert$	$E[\lvert r_i^{c_j}\rvert]$	$\texttt{med}(\lvert r_i^{c_j}\rvert)$	$\sigma(\lvert r_i^{c_j}\rvert)$	$\lvert\{r_i \lvert\lvert r_i\rvert \geq 150\}\rvert$
Africa	25	232	202	105	21
China	2260	64	52	54	107
Iran	64	435	185	1027[1]	47
Turkey	1695	412	415	249	1328

Table 4.1: Basic Features - African, Chinese, Persian, and Turkish music time series.

To determine the minimum length of analysis permissible, a small subset of the compositions were chosen at random from my Western analysis in Chapter 3. Each series, r_i, was then segmented into a set of sub-series of 52 duration elements, $L = 52$, and fractal exponents were computed for $r_i(1 : L), r_i(1 : 2L), \ldots, r_i(1 : |r_i|)$ and $r_i(1 : L), r_i(L+1 : 2L), \ldots, r_i(kL + 1 : |r_i|)$. The results showed that although the fractal exponents vary slightly, when the sub-series is sufficiently long (e.g., 150 duration points), the fractal exponents of the sub-series merge to the value for the entire series. Figure 4.1 shows an example of this analysis for a representative composition by Grieg.



(a) DFA exponents at incremental lengths: $\alpha_{r(1:L)}, \alpha_{r(1:2L)}, \ldots, \alpha_{r(1:l)}$. The mean α value is shown in red.



(b) DFA exponents of segments: $\alpha_{r(1:L)}, \alpha_{r(L+1:2L)}, \ldots$. The mean α value is shown in red.

Figure 4.1: DFA exponents are computed for different segmentat“little bird” of a composition by Grieg called “little bird” (“Lyric Pieces, Op. 43 No. 4. in D minor”). The total length of this music time series is $l = 340$, and the segmentation length visualized here is $L = 52$.

Accordingly, the minimum length of analysis for my non-Western collection is $L_{\texttt{min}} = 150$ and any series shorter than $L_{\texttt{min}}$ were excluded. Table 4.1, presents general statistics on the length of time series analyzed for each region. It should also be noted that although the following methodology enables a novel structural analysis of the latent temporal information in non-Western music from Africa, China, Iran and Turkey, the use of Western musical notation may not be ideally suited for capturing the richness of music in this context; transcriptions of deep interpretations of non-Western music may be impossible [ATTB91, Tou05, TKSW07, MCL+07, Şen11].

4.3 Non-Western Temporal Power Law Exponents

For this collection of over 1500 non-Western music time series, fluctuation structures in the musical rhythms decrease exponentially with $0.5 \leq \alpha \leq 1$. Figure 4.2 presents the median α values for the four regions of analysis: Africa, China, Iran and Turkey. All four regions have median DFA exponent values close to 0.5: $\alpha_{\texttt{Africa}} = 0.41, \alpha_{\texttt{China}} = 0.41, \alpha_{\texttt{Iran}} = 0.47$, and $\alpha_{\texttt{Turkey}} = 0.48$. These low values of α are indicative of lower predictability in the rhythmic structure of music from these regions. Table 4.2 provides a more granular analysis for values of α, grouped by region. The Turkish collection, despite its larger size, does not show much variation ($\sigma_\alpha = 0.13, \sigma_H = 0.14$).

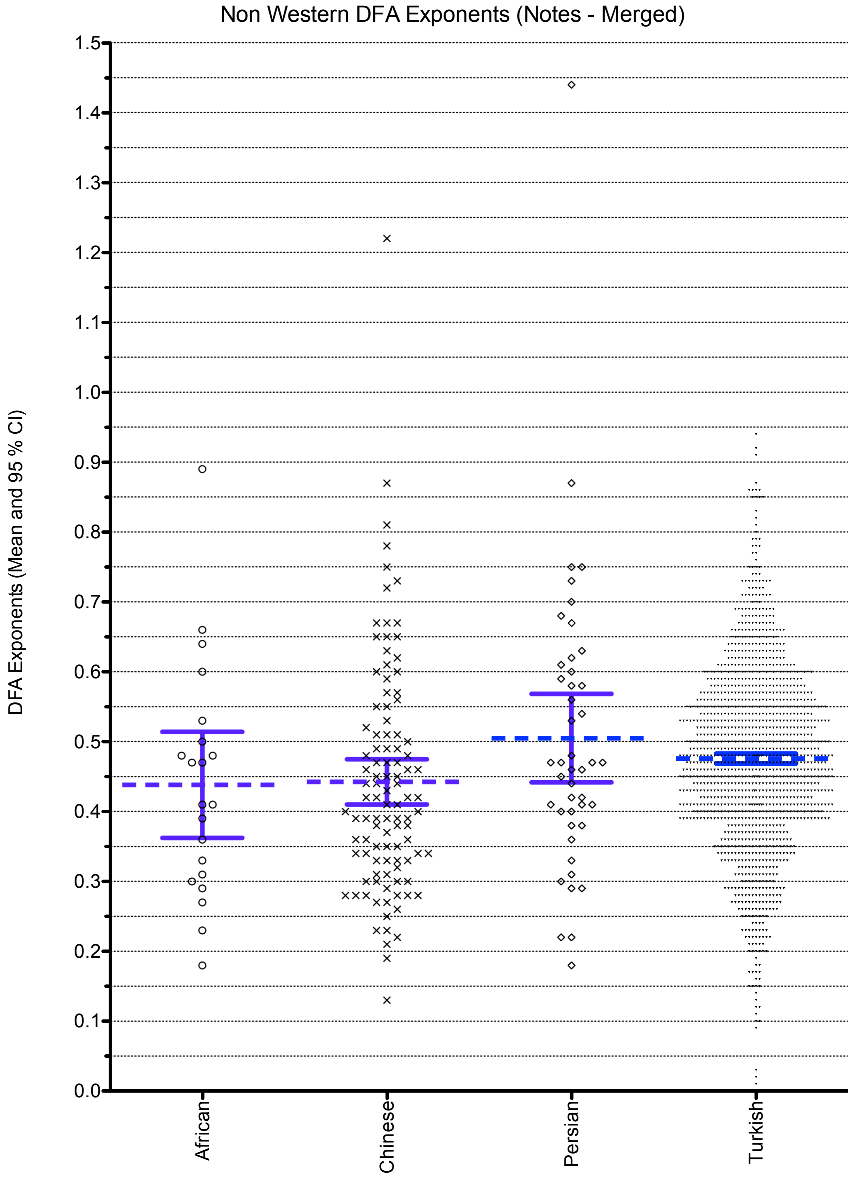


Figure 4.2: Detrended Fluctuation Exponent (α) for compositions in $R_{\texttt{African}}, R_{\texttt{China}}, R_{\texttt{Iran}}, R_{\texttt{Turkey}}$ sub-categorizations of the $R_{\texttt{non-Western}}$ collection. The Mean and 95% confidence intervals are marked in blue.

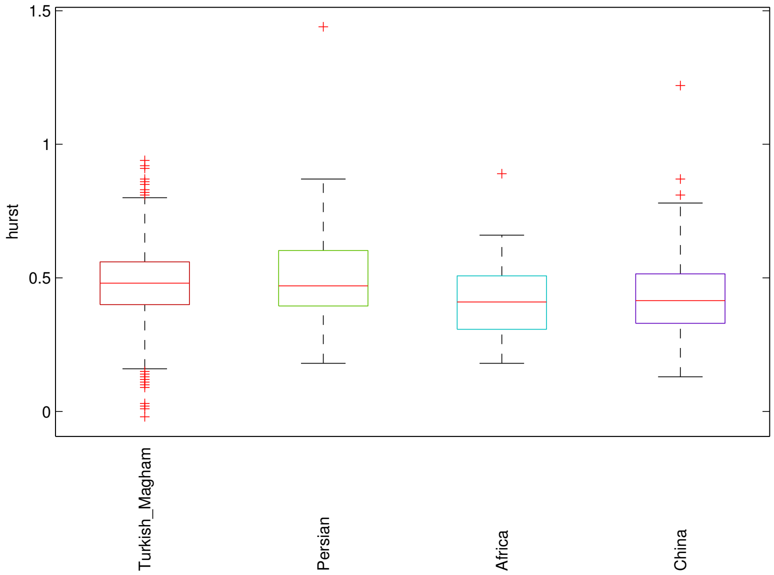


Figure 4.3: Hurst Exponents (H). Median of H for compositions in each collection, $R_{\texttt{African}}, R_{\texttt{China}}, R_{\texttt{Iran}}, R_{\texttt{Turkey}}$, are shown in red in each box. The lower and the upper boundaries of the boxes represent 25% and 75% interquartiles of H. Anomalies are marked in red '+'.

Finally, in all four non-Western groups considered, power law correlation exponents, α (Figure 4.2) and H (Figure 4.3) indicate high structural unpredictability. That is, duration time series in these non-Western compositions have less repetitive, long-range rhythmic structures, and change more often.

Table 4.2: Non-Western Music - Hurst (H) and DFA exponents (α).

		Mean	Median	Standard Deviation	Coefficient of Variation
$\lvert R_{\texttt{Africa}}\rvert = 21$	α	0.44	0.41	0.17	0.38
	H	0.56	0.57	0.12	0.22
$\lvert R_{\texttt{China}}\rvert = 107$	α	0.44	0.41	0.17	0.37
	H	0.54	0.51	0.17	0.31
$\lvert R_{\texttt{Iran}}\rvert = 47$	α	0.51	0.47	0.21	0.42
	H	0.61	0.60	0.15	0.25
$\lvert R_{\texttt{Turkey}}\rvert = 1328$	α	0.48	0.48	0.13	0.28
	H	0.57	0.58	0.14	0.25

The collection's mean H values (Table 4.2) are close to 0.5: persistent long-range similarities are absent. Finally, higher average values of Hurst exponents in the Persian collection (Table 4.2) indicate a rhythmic pattern of composition: an increase in a Persian duration time series is more likely to be followed by other increases.

4.4 Binary Classification of non-Western Compositions

This section demonstrates the significance of temporal self-similarity power law exponents as descriptors (*signatures*) of the African, Chinese, Persian, and Turkish music samples available in $R_{\texttt{non-Western}}$ (Section 4.2.2). In this section, I choose the name of each region or country as a *composer* identity; this categorization assumes a cohesive musical tradition within a culture or region [Per06, GH08]. I use binary classification as outlined in Section 2.4.4; all pairs of the non-Western composition groupings are classified using logistic regression (linear classifier used here) and J48 decision trees (non-linear binary classifier used here) implemented in WEKA [HFH$^+$09]. The classifiers train on fractal exponents and predictability features listed in Table 2.4.4. The accuracy of these classifications depends on the amount of information latent in this temporal representation of structural repetitions in non-Western music. A 10-fold cross-validation is used for both classifiers, and unless otherwise specified, no other filtering has been applied. When classifying two sets of non-Western compositions that differ in size by more than a factor of 2.5, the larger set has been sub-sampled randomly. Lastly, mean absolute error (MAE), weighted F-measures and confusion matrices (Section 2.4.4) are used to assess classification accuracy.

More precisely, structural information extracted from the temporal representations of this study's African, Chinese, Persian and Turkish music pieces are used for binary classification. The resulting classification accuracies signify the information-rich structural identity of music , included in this dissertation.

The classification results for the African, Chinese, Persian and Turkish music pieces are summarized in Figure 4.4. In this context, the African music pieces, $R_{\texttt{Africa}}$, are significant because they are the most difficult to distinguish and classify. That is, using fractal and predictability exponents, the African compositions yield the lowest classification accuracy; while the best accuracy results were observed in comparing Chinese versus

Turkish (91%) and Persian versus Turkish (86%) compositions available in $R_{\texttt{nonWestern}}$.

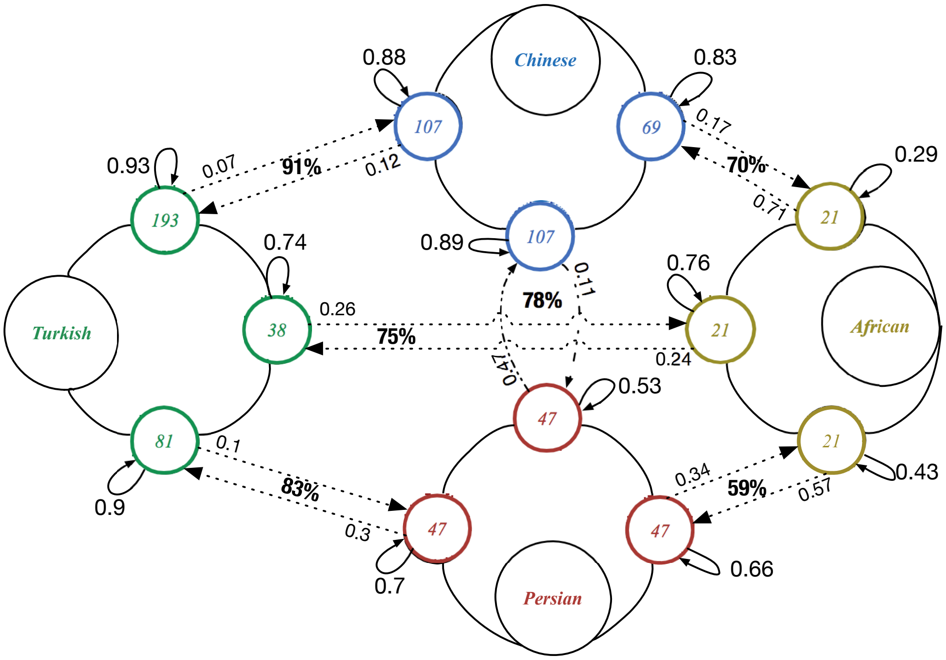


Figure 4.4: Binary Classification (Decision trees) for the $R_{\texttt{Africa}}$, $R_{\texttt{China}}$, $R_{\texttt{Iran}}$ and $R_{\texttt{Turkey}}$ music collections. All possible pairings are displayed. Numerical values inside circles indicate the number of instances considered; an edge value, e_ij, denotes the percentage of instances with label i that were labeled as j; self-loops represent the percentage of instances correctly classified; and the classification accuracy percentage is marked in bold, black in between edges.

To better understand the influence of various features included in the classification (Table 2.4.4), detailed break-down of various features and the resulting classification accuracies are presented in Appendix A (Tables A.1, A.3, A.5, and A.9).

Among all non-Western classification pairs considered, African pieces resulted in lowest true positives (Tables A.1, A.3, and A.5 in Appendix A), and can be considered most difficult to be captured by temporal power law and predictability exponents. However, even in this case, classification achieved accuracy results of at least 70%. Classification accuracy was highest between Chinese and Turkish pieces, 91% (Table A.9). The ability to linearly discriminate between the two collections indicates significant disparities between their rhythmic fractal features. Finally, my analysis shows a clear distinction between Persian and Turkish compositions. In this case, both classifiers achieved higher than 80% accuracy. Also, fractal exponents produced more true positives for Turkish pieces. The ability to distinguish between music from two geographically-approximate regions is significant on its own as it highlights subtle cultural and musical differences between the two cultures.

4.5 Western vs. Non-Western Classification

After classifying all possible binary pairings of the sub-categories in this non-Western collection, $R_{\texttt{nonWestern}}$, (Figure 4.4), this section studies the efficacy of Western vs. non-Western binary classifications. That is, this section investigages the success of temporal power law exponents in binary classifications of music samples from $R_{\texttt{nonWestern}}$ and $R_{\texttt{nonWestern}}$.

The feature set used consists of fractal and predictability exponents (Table 2.4.4) for the following two categories:

- **Western:** Grieg, Gershwin, Mozart and Scarlatti compositions from $R_{\texttt{Western}}$, and

- **Non-Western:** African, Chinese and Persian music from $R_{\texttt{non-Western}}$.

Instances which corresponded to lengths less than 200 in the Western case, and 150 in the non-Western, were not used in the classification. The results, shown in Tables 4.3 and 4.4, demonstrate 90.3% accuracy. Binary classification using only DFA exponents, α, as features, still yields accuracy greater than 82% for both logistic regression and decision trees. This suggests that the existing temporal power law features in Western and non-Western music are distinctly different.

Table 4.3: Binary classification results using logistic regression: Western vs. Non-Western categories.

$\lvert R_{\texttt{Western}}\rvert = 249$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Non-Western}}\rvert = 175$	Classified (%)	Absolute Error	F-Measure	Non-Western	Western	
Fractal Exponents, Π	90.3	0.11	0.9	154	21	Non-Western
				20	229	Western
Π	89.2	0.14	0.89	157	18	Non-Western
				28	221	Western
Fractal Exponents	82.5	0.25	0.83	140	35	Non-Western
				39	210	Western
α	82.1	0.26	0.82	137	38	Non-Western
				38	211	Western

For both classifiers — logistic regression (Table 4.3) and decision trees (Table 4.4) — benefit from combining predictability exponents, Π, with fractal exponents. However, fractal exponents on their own achieve considerably accurate — greater than 80% — classification results as well.

Table 4.4: Binary classification results using decision trees (J48): Western vs. Non-Western categories.

$\lvert R_{\texttt{Western}}\rvert = 249$ $\lvert R_{\texttt{Non-Western}}\rvert = 175$	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
				Non-Western	Western	
Fractal Exponents, Π	90.3	0.1	0.9	155	20	Non-Western
				21	228	Western
Π	88.2	0.14	0.88	150	25	Non-Western
				25	224	Western
Fractal Exponents	81.4	0.23	0.82	147	28	Non-Western
				51	198	Western
α	83	0.26	0.83	160	15	Non-Western
				57	192	Western

In both classification methods, fractal exponents seem to result in more true positives in Western compositions, which can be interpreted as more discriminative scale-free structural characteristics in such compositions. This demonstrates that the temporal power law features of this non-Western collection are more homogenous than their Western counterparts.

4.6 Conclusion and Future Works

In this chapter, I computed temporal power law correlation exponents (fractal exponents) for symbolic music categorized based on geographical origin. This collection consisted of symbolic music from Africa [Sap05], China [Sap05], Iran [Abd11] and Turkey [Kar12]. These music pieces form the non-Western collection of this dissertation; the choices included in this non-Western collection are limited to availability of non-Western symbolic music, at the time this research was undertaken; the collection is further sub-categorized by geographical labels: Africa, China, Iran and Turkey. In this temporal analysis, over 1500 music time series were included. For each music time series, I computed fractal exponents, and for each region, the Detrended Fluctuation analysis exponents, α, were reported (Figure 4.2) and discussed. Generated fractal and predictability Exponents (Table 2.4.4) were used to distinguish between pairs of non-Western regions in this context (binary classification). This investigation showed that such classifications, using self-similarity information in musical rhythm, are highly accurate (Figure 4.4). Also, I used these exponents for the classification of Western versus non-Western music pieces; that is, the binary classifier was provided access to all compositions attributed to Gershwin, Grieg, Mozart, and Scarlatti available in the collection of Section 3.1 (collectively labelled as *Western*) and African, Chinese and Persian music pieces available in this dissertation (collectively labelled as *non-Western*). Throughout all classification analyses, I applied binary classification using logistic regression and decision trees with 10-fold cross validations.

This analysis demonstrated the manifestation of the power law phenomenon — temporal power law correlation exponents component for the African, Chinese, Persian and Turkish samples available fall in $\approx (0.5, 1.5)$ range as shown in Figure 4.2 — for these non-Western compositions as well.

This analysis shows that not only are power law structural features not unique to Western compositions (Section 3.2), but that also such features have sufficient information content to be used as identifiers (Section 4.4).

My investigation of this music collection shows that the power law phenomenon, which is well-studied for music compositions from Europe and North America (i.e., compositions often referred to as Western classical compositions) [Zip49, VC75, VC78, GTM95, PB00, MVW^{+}03, LCM12], is consistently present in music samples of African, Chinese, Persian and Turkish music analyzed here as well. Given the ubiquity of power laws in nature [WS90], it is not surprising to find temporal power laws manifest in music from various regions in the World. In other words, the emergence of fractal power laws and the culture-dependent granularity in this collection of non-Western music (Figure 4.2), may be used in support of the notion of "universality in diversity" [DM77] (Section 4.1). The existence of fractal exponents and their successful application as identifying signatures in classification demonstrated that these attributes of music's structural regularity are computationally significant. These efficiently computable signatures can be used for large-scale music information retrieval tools. More precisely, structural similarities, represented by fractal exponents, will ultimately be critical for a global and culture-independent, music search engine. The high accuracy classification results of these exponents show that classification techniques applied to Western classical music may also be applied to other cultures and can further be used in comparative analysis of structural style. Finally, my results present fractal exponents as promising parameters used in a uniform comparative analysis of music from different regions, traditions or eras.

5

mCaptcha: A New Challenge-Response Security Test Based on Music

The Turing test, first proposed by Alan Turing [Tur50], is an attempt to distinguish machines from humans. A computationally efficient realization of such a Turing test is a CAPTCHA — Completely Automated Public Turing Test to tell Computers and Humans Apart — developed by Luis von Ahn at Carnegie Mellon in 2000 [vABHL03]. The cornerstone idea of CAPTCHAs is to take advantage of certain features of human cognition to solve problems that are easy for humans but are currently too computationally expensive or intractable to be solved by computers. Since their introduction, CAPTCHAs have been widely adopted as a practical means of access control and security against automated computer attacks (bots) and spam for major e-commerce, social networks, and online posting sites (e.g., Wikipedia). Other applications include protecting the integrity of registration processes (e.g., e-mail service providers such as Hotmail, Gmail, Yahoo!) and online polling [cap, vABL04, gov12, vABHL03]. Their seminal design and ease of integration into web interfaces and applications have led to their ubiquity online. Though the seminal work of von Ahn et al. [vABHL03] led to the widespread use of such tests, they also note prior similar work [BvALH00, vABHL03]. The first mention of systems or notions similar to CAPTCHAs are in an unpublished manuscript by Naor [Nao96] and a system developed by Lillibridge et al. at AltaVista [LABB01]. However, von Ahn et al. presented alternative varieties for such tests in the CAPTCHA system [BvALH00, vABL02, vABL04], highlighted benefits of adopting an empirical notion of security in this context [vABHL03] and facilitated the large-scale deployment of such tests [vAMM$^+$08]. Finally, a modification of CAPTCHAs — reCAPTCHA [vABM08] which has helped to successfully digitized millions of old books [vAMM$^+$08] — demonstrates the potential power of leveraging human time and processing power for the advancement of computationally expensive, if not difficult, problems in many research fields.

This chapter proposes a novel design that uses the framework of existing CAPTCHAs. My proposal integrates music into its design components, relies on characteristics of human musical cognition for its improved usability and takes advantage of the current computational difficulty of music information retrieval and scene analysis algorithms for its enhanced security. Music CAPTCHAs (mCaptchas)[1] rely on particular features of music cognition to generate efficient human-machine distinguishability tests. The objective of this design is to enhance the existing computational Turing tests by improving user experience (usability), amplifying the computational gap between machines and humans (security), and taking advantage of the available human computational resource for crowdsourcing oriented towards open problems dependent on large-scale user-generated music information.

[1] The system has a provisional patent filed on November 6, 2012 as “Music CAPTCHA System and Method”, US patent application/PCT international application number 61722848.

5.1 Problem Statement and Relevant Research

The problem of designing a CAPTCHA can be described as follows. The entity accessing a web site (the "user," and which may be a bot or a human), requests access and is presented with a challenge. The user is said to *pass* the CAPTCHA — is identified as human by the system — if and only if his response to the challenge is correct. The challenge is a question with a unique answer, and it is designed such that it is relatively easy for humans but computationally difficult for the state-of-the-art programs.

A typical CAPTCHA will present the user with a distorted word on a noisy background, and will challenge the user to detect and type the correct phrase. Given the ubiquity of CAPTCHAs and usability issues with the original text-based form, various forms of CAPTCHAs have been devised. In particular, audio CAPTCHAs were created to improve accessibility of these tests to individuals with visual impairments and also to provide an alternative form for when the user finds reading the distorted CAPTCHA words difficult [TSHvA08].

5.1.1 Existing Audio CAPTCHAs

Audio CAPTCHAs consist of a set of scrambled (English) letters and numbers to be identified by a user from a background of noise. Audio CAPTCHAs were created to extend the accessibility of textual CAPTCHAs to individuals with visual impairment. An audio CAPTCHA is created by overlaying voices of different speakers pronouncing (random) letters or numbers on top of noise. The user must correctly identify the digits or characters spoken to pass the test. Although such designs are important in that they make sites such as Google and Digg available to the visually impaired, they have been compromised by (computational) adversaries, computer algorithms which use various machine learning techniques to learn the pronounced letters and numbers from the audio. The security of audio CAPTCHAs was first tested and shown to be compromised for [TSHvA08] three types of audio CAPTCHAs used in Google.com, Digg.com and reCaptcha.net. Tam et. al., used machine learning techniques to segment, classify components and ultimately break visual CAPTCHAs [SC04]. Furthermore, Tam et. al. [TSHvA08] go on to suggest the use of meaningful phrases in place of randomly placed, random letters or numbers over a background noise. During the design of CAPTCHAs, a balance must be struck between the computational difficulty of solving a CAPTCHA and the human usability. This human element is the second pillar of designing CAPTCHAs that needs to be given adequate treatment. Recent research has shown that CAPTCHAs are often difficult for humans to solve [BBF$^+$10]. Moreover, Bursztein et al. [BBF$^+$10] showed that non-musical audio CAPTCHAs are problematic in that not only users have difficulties in solving them, but also that approximately only a third of the participants concur on their responses to the same CAPTCHA challenge.

5.1.2 Distinct Features of CAPTCHAs

> *"If you tell me precisely what it is a machine cannot do, then I can always make a machine which will do just that."*
>
> John von Neumann [Wikb]

Any CAPTCHA scheme has three significant design tenets [Nao96, vABHL03]:

- First, generating CAPTCHA instances should be computationally practical (*efficiency*). Otherwise, a large-scale adoption of the scheme would be impractical. The generation algorithm uses an AI (Artificial Intelligence) that, by consensus, is unsolved and whose approximate solutions need much computational resources, time or space [vABHL03],
- Second, the scheme should ensure that the generated CAPTCHAs are easy for humans to solve (*usability*),
- Third, answering the challenge should require much more computational resources (e.g., time) for the state-of-the-art algorithms (*security*).

These tenets will form a framework of design and evaluation for mCaptchas proposed in Section 5.2.

5.1.2.1 Usability of CAPTCHAs

Though exact parameters were never specified, for practicality measures, a large portion of the human population should be able to answer the challenge correctly in a very short period of time [vABHL03].

5.1.2.2 Security of CAPTCHAs

> *"The enemy knows the system. One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them."*
>
> Claude E. Shannon [Sha49]

Modern cryptography studies the notion of computational security in an adversarial model. It redefines security not in the impossibility of a system being broken, but rather in the infeasibility of a computationally-bounded[2] adversary learning more than what is publicly known [GM82]. Security is *not* based on the secrecy of a component or code in the system (Kerckhoffs's principle [Ker83]).

In the context of multimedia security, protected content should only *appear* secure to a computationally bounded security (e.g., security through scrambling images or video) [Lia08]. Security of CAPTCHAs is empirical; it is based on a consensus of computational difficulty and not on proven intractability [Nao96, vABHL03]. A CAPTCHA is considered to be secure if an adversary — having access to the state-of-art solutions, algorithms and systems — cannot correctly and efficiently solve a problem that is easy for humans. More precisely, an adversary, in this context, is assumed to be any automated design or algorithm which has access to the state-of-the-art solutions to the underlying difficult problem used for CAPTCHA challenges. An adversary knows how the system generates CAPTCHAs, but does not have access to the parameter choices or the randomly chosen input, determining the correct response to the challenge. It should also be noted that unlike many cryptographic protocols, a particular CAPTCHA scheme considered to be secure today might not be so in the future [vABHL03]. As state-of-the-art AI algorithms improve over time, so should CAPTCHA schemes.

As an example of empirical *proof* of security, consider reCaptchas. This scheme was introduced after the textual CAPTCHA scheme [BvALH00, vABL02, vABL04] was

[2] Saying that the adversary's computational power is bounded means that the adversary runs in polynomial time. The adversary's success probability in breaking the system, given what is already known about the system and the input, is negligible though not zero. This turns into the adversary from an all powerful adversary to a realistically pragmatic one.

shown to be susceptible to pattern recognition attacks by Mori and Malik [MM03]. The scheme uses actual English pairs of words, in place of randomly chosen alphanumeric letters, and harnesses "wasted" human computation to digitize millions of books [vAMM$^+$08, vA05]. In this context, von Ahn et al. argue for the scheme's security empirically[vAMM$^+$08]:

> Because computer programs can easily attempt to pass the CAPTCHA multiple times, if a computer has a success rate of even 5%, the CAPTCHA is considered broken. A typical convention is that a program should not be able to pass the CAPTCHA with a success rate of more than 1 in $10,000$. (Downloading $10,000$ CAPTCHA images requires substantial usage of bandwidth, exposing the IP address as potentially abusive.) Our system uses more than $100,000$ words, which yields a probability of random guessing that is much smaller than $1/10,000$. By contrast, conventional CAPTCHAs that use seven random characters yield an even smaller probability of success for random guessing: $1/367$.

This probability ($i.e.$, $1/367$) corresponds to randomly guessing meaningful English phrases. In Section 5.2, I introduce mCaptchas as an alternative to existing noise-based CAPTCHAs. This scheme uses the complexity of music to improve security and usability. It uses latent information in music and unique attributes of human auditory perception; its security relies on the disparity between capabilities of the state-of-the-art music identification algorithms and humans to distinguish two music streams from a mixture played simultaneously, and to answer contextual challenges.

5.2 mCaptcha: A Music-Based Computational Turing Test

A Turing test is an interactive dialogue: a verifier, V, presents a challenge , c, to a requester , A,, A responds with a solution to the challenge, s_c^A, and V *deems* A to be human after verifying s_c^A. Turing proposed to use a human verifier [Tur50]; whereas, von Ahn et al. [vABL02, vAMM$^+$08, vABHL03, vABL04] introduced the notion of *computational Turing tests*— computational verifiers distinguishing between polynomial-time adversaries and humans. The system proposed in my thesis differs from the computational framework of CAPTCHAs in that it:

- uses music: a complex and information-rich primitive,
- uses a slightly modified notion of CAPTCHA-security:
  - security is based on the computational difficulty of distinguishing between two *valid* streams, and
  - security is dependent on contextual questions about music.

Here, the user is presented with a single audio stream which can be heard as two streams by humans, but which *appears* as a single, inseparable unit of music for the current state-of-the-art audio identifiers. More precisely, the mCaptcha system presents a requester, R, with a pair consisting of a composite stream and a corresponding contextual challenge: (m_j, c_{m_j}). The stream is constructed from a widely known piece of music (the primary, p_{m_j}) which is then segmented and intertwined with segments from a distracter (a secondary music stream, s_{m_j}, from a very different genre (e.g., the primary might be rock and the secondary might be classical)). The requester responds with a *solution* to the challenge, $r_{c_j}^R$.

A requester is said to have *passed* the test (i.e., the sequence has terminated in `accept`) if a correct response to the posed challenge has been received before the mCaptcha expires.

In a proof-of-concept implementation of this scheme (Section 5.3), the primary and secondary music snippets — a few seconds long representing the most recognizable portions — are chosen from different decades and genres of western music to be identifiable by members of the potential user pool. However, this choice is arbitrary and the design is not limited to any particular genre, language or style of music. In fact, one of the advantages of the proposed system is possible future customization to a particular user's music library — assuming that the collection is large enough and access is granted. The challenge chosen should have an unambiguous answer.

5.2.1 Components and Design Considerations

The architecture of the mCaptcha system comprises three components:

1. the *primary-secondary stream selection* module,
2. the *generator* module, and
3. the *challenge* module.

I give a brief overview of these modules in this section, and highlight key design considerations. I describe two particular implementations of these modules in more detail in Section 5.3 and viability analysis (security and usability) at greater depth in Sections 5.4 and 5.5.

5.2.1.1 Design Guidelines

The key design considerations of the mCaptcha scheme are based on the following CAPTCHA tenets [Nao96, vABL02, vABHL03, vABL04]. An mCaptcha instance is a tuple, $(j, c_j, \lambda(c_j))$, consisting of a composite music stream, referred to as the context (j), the challenge associated with this context (c_j) and a corresponding unambiguous response to the challenge $(\lambda(c_j))$. The scheme accepts human responses, r_H, and rejects those of any adversary, A: That is, $\langle(j, c_j, \lambda(c_j)), r_H\rangle = \texttt{accept}$ and $\langle(j, c_j, \lambda(c_j)), r_{\texttt{A}}\rangle = \texttt{reject}$, with high probability. An assumption of the mCaptcha scheme is that users can hear the generated music stream. In other words, similar to the CAPTCHA model, the scheme should be usable by as large a proportion of the population as possible. [vABHL03]

1. **Efficiency of instance generation:** A generator, $G(.)$, efficiently generates mCaptcha music streams, together with their corresponding challenge and the correct solution (Figure 5.1). The generator requests and receives a random song from the primary collection. Next, it sends a request and receives a secondary song. The secondary song may be chosen at random, or it may be chosen adaptively. In the latter case, efficiently-computable features of the primary, such as regularity in its temporal structure, are used to select a secondary. Finally, G calls the challenge module to generate a challenge, response pair. The challenge should be contextual and dependent only on the primary stream. The corresponding correct response to the challenge should be unambiguous. Detailed description of the generator is provided in Section 5.3.

2. **Ease of use by humans (usability):** Humans should be able to answer the challenge correctly, efficiently and with ease. The scheme should not be overly complicated or annoying to discourage or prevent human users from solving the challenge (user experience). Moreover, the challenge should be chosen such that a typical human user can accurately respond to it (accuracy). The type of contextual information needed in the challenge determines the amount of time needed to respond (efficiency) and the pre-determined time-out period. I discuss mCaptcha's usability by providing intuition for its improvement over existing schemes, and provide usability tests, $T_H(.)$, for the construction in Section 5.5

3. **Computational difficulty for an adversary (security):** A computationally-bounded adversary, with access to the state-of-the-art algorithms, should fail to correctly answer the challenge on almost all instances [Nao96, vABHL03]. Naor [Nao96] further adds that this probability of failure can be amplified, to an almost certainty, by requiring the requester to solve *multiple*[3] instances correctly. In Section 5.4.1, I provide evidence for this scheme's enhanced security, and present a security test, $T_S(.)$, for my constructions of Section 5.3.

4. **Succinct instance representation:** For integration into a web interface, a generated mCaptcha instance should be efficiently (in time and space) delivered, possibly stored until requested, and represented to the user. A practical, large-scale implementation of such an interactive system requires that the instances representation can be efficiently communicated and stored.

[3] From a usability point of view, this leads to an interesting question: after solving how many CAPTCHAs will a typical user give up.

In Section 5.3, I discuss the influence of these design requirements on my two particular proof-of-idea implementations at greater depth.

5.2.1.2 Architecture

The proposed scheme is an interactive system: a system involving a sequence of challenge-response communications from and to the user. The challenge-response sequence starts with the arrival of a request from a user, who may be a bot or a human. Consequently, the user's request is countered by a music stream and an associated challenge. The user must submit a response to the challenge within a pre-set, appropriate short period of time. If no response is received, the system rejects this round. Otherwise, the input response is verified against the stored, correct answer for this instance. The user passes the mCaptcha test if and only if his response matches the correct answer. The streams used in building the mCaptcha go through modules which modify their underlying structure. These modifications can be thought of — loosely speaking — as evolution phases. I highlight each such phase in the following, and note those which appear computationally-irreversible to an adversary.

More precisely, the architecture of this system consists of a primary-secondary selection, a generator and a challenge selection module. The mCaptcha challenge-response sequence begins with the arrival of a request from a user, which may be a bot or a human. The requester is presented with a single music stream, m_{p_r,s_r} and an associated challenge, $c_{m_{p_r,s_r}}$; the composite stream m_{p_r,s_r} can be easily heard as two distinct music streams, p_r and s_r, by humans, but *appears* as a single inseparable unit of music to the adversary. The user must submit a response, r_u, to this challenge within a predetermined short period of time, t^*. If no response is received within t^* seconds, the system rejects this round and refreshes to another randomly selected mCaptcha; otherwise, the response is verified to match the corresponding correct answer, $\lambda(c_{m_{p_r,s_r}})$, stored for this instance. The requester *passes* the mCaptcha test if and only if his response matches, or almost matches, the correct answer: That is, $\lambda(c_{m_{p_r,s_r}}) \approx r_u$. The three modules used to generate (m_{p_r,s_r}, $c_{m_{p_r,s_r}}$, $\lambda(c_{m_{p_r,s_r}})$) are:

1. **Primary-Secondary Stream Selection.** In this module, a primary (P) and a secondary (S) stream selected to be combined into a single auditory stream: the mCaptcha stream (m). Each stream will be represented as a time series of amplitude values, or amplitude values mapped to a particular range such as $(-1, 1)$. Since the human auditory system has a limited hearing range [Ols67], time series representation of music pieces will also have a limited range of possible values.

   (a) *Primary selection.* The primary stream, $P = \{p_i\}_{i=1}^n$, is the most significant auditory component of m since the mCaptcha's challenge depends on P.

   (b) *Secondary selection.* The secondary stream, $S = \{s_j\}_{j=1}^m$, is used as a distractor, and its choice influences both security and usability. A successful mCaptcha strikes a balance between similarity and disparity of S and P: (1) S should be sufficiently different from P for the user to easily distinguish between the two (usability), and (2) the chosen secondary should be adequately similar to P in musical quality and structure (e.g., rhythm and spectral signature) so that the adversarial separation between the two is difficult (security). In other words, such structural similarity improves security through *computational indistinguishability* between two valid music streams.

2. **mCaptcha Generator.**

   (a) *Segmentation*. This sub-module segments both primary and secondary streams into blocks of random length[4]:

   - $\texttt{Seg}_{r \leftarrow U_*}(P) = [\langle P^1 \rangle \langle P^2 \rangle \ldots \langle P^{n_p} \rangle]$,
   - $\texttt{Seg}_{r \leftarrow U_*}(S) = [\langle S^1 \rangle \langle S^2 \rangle \ldots \langle S^{n_s} \rangle]$.

   The segmentation function, Seg, uses its internal random choices — denoted by r and which is chosen uniformly at random — to divide the audio stream into blocks of random length. The total number of blocks for the primary and the secondary are denoted by n_p and n_s respectively. The segmented blocks, especially for the primary, cannot be too short or too long. If the generated blocks are too short, human usability suffers. At the other extreme, although very long blocks may be aesthetically more pleasing, increase the likelihood of audio identification by the adversary. Note that no irreversible modification has yet been applied.

   (b) *Stochastic linear transformations*. This sub-module, combines blocks of primaries and secondaries by dropping certain blocks at random and interleaving the primary and secondary blocks. A block, $\langle P^i \rangle$, being *"dropped"* refers to multiplying values of that block by zero. This sub-module is the first evolution phase which inserts extra information — in this case randomly chosen weights

[4] For simplicity, streams are represented in capital letters (e.g., P) with their corresponding values over time in lower case (e.g., $P = [p_1 \ldots p_n]$). Segmented blocks of a particular stream are represented by the name of the stream and the index of the block. For instance, $\langle P^i \rangle = [p^{i,1}, \ldots, p^{i,l}]$ corresponds to the $i^{\texttt{th}}$ block, of length l, taken from audio stream P.

— into the construction. Each stream is multiplied by weights chosen uniformly at random from the set of possible weights $W_1 = \{0, \epsilon_1, \dots, \epsilon_j, \dots, 1\}$. This set includes zero, resulting in a particular block being "dropped", and 1 which corresponds to the no change case. Other weights, ϵs, are chosen close to one (e.g $0.92 \leq \epsilon_j \leq 0.98$). I denote the stochastic linear transformation by SLT, and further distinguish between its treatment of the primary and secondary streams for the sake of clarity.

Let w_i^1 be the $i^{\mathtt{th}}$ weight chosen from W_1 uniformly at random: $w_i^1 \leftarrow_U W_1$. In all cases, $\leftarrow_U$ represents sampling uniformly at random. I use $w_i^1\langle P^i\rangle$ to mean a multiplication of all values in block P^i by weight w_i^1. Then the stochastic linear transformation for each stream can be written as:

- $\mathtt{SLT}_{\mathtt{W_1,r\in U_*}}(\mathtt{Seg}_{r\leftarrow U_*}(P)) := [w_1^1\langle P^1\rangle \dots w_{n_p}^1\langle P^{n_p}\rangle]$,

  where $\forall i, 1 \leq i \leq n_p, w_i^1 \leftarrow_U W_1$
- $\mathtt{SLT}_{\mathtt{W_1,r\in U_*}}(\mathtt{Seg}_{r\leftarrow U_*}(S)) := [\langle w_1^1\langle S^1\rangle + \langle r^1\rangle\rangle \dots \langle w_{n_s}^1\langle S^{n_s}\rangle + \langle r^{n_s}\rangle\rangle]$

  where $\forall i, 1 \leq i \leq n_s, w_i^1 \leftarrow_U W_1$, and

  $\langle r^i\rangle$ is noise with the same length as S^i ($|\langle S^i\rangle|$).

In very broad terms, this transformation takes in the segmented primary and secondary, chooses fillers of random length to be inserted between various primary blocks, randomly drops some of the primary blocks, and fills in the "gaps" — which have random lengths from a particular range ensuring that the blocks are not too short or too long — with modified blocks of the segmented secondary. Stochastic linear transformations of the secondary blocks may map it to some fractional noise — white (Figure 2.1), brown (Figure 2.2) or pink noise (Figure 2.3)). Let's denote the resulting streams from this evolution by $\tilde{P}$ and $\tilde{S}$. To an adversary who does not have access to the random weights, *exact* recovery of P and S is not possible.

(c) *Stochastic combination.* The stochastic combination module, `Glue`, builds a new stream, $\hat{m}$, from "interleaving" $\tilde{P}$ with $\tilde{S}$:

- $\texttt{Glue}_{W_1,W_2,W_3}(\tilde{P},\tilde{S}) := [\langle w_1^2 \tilde{P^1} + w_1^3 \tilde{S^1} \rangle \ldots \langle w_{n_p}^2 \tilde{P^{n_p}} + w_{n_p}^3 \tilde{S^{n_p}} \rangle]$
  where $\forall i, 1 \leq i \leq n_p, 1 \leq j \leq 3$, $w_i^j \leftarrow_U W_j$.

The weights — chosen uniformly at random from W_1, W_2, W_3 — are heuristic design parameters and will emulate certain blocks being dropped at random, and blocks being added together. Particular instances of these parameters will be discussed in the prototype mCaptcha implementation of Section 5.3.

(d) *Global modification.* Finally, a transformation that alters global characteristics of the mixture, such as its temporal structure, is applied. Permissible audio manipulations include only transformations which cause either inaudible or minimal changes. The resulting stream is the output mCaptcha stream, m.

3. **Challenge Selection.** This module outputs a contextual question. The correct answer to the challenge depends on the primary, P, and is unambiguous. The contextual questions rely on real-world knowledge that is easy for humans to acquire and non-trivial for computers. The challenge has a unique answer; however *approximately* correct responses are accepted. This *fuzziness* is not inherent in the question, but rather in the shortcomings of humans in recalling certain information (e.g., name of an artist or a song) exactly. Lastly, the question should not be subjective, and so

candidate challenges involving music genre should not be considered.

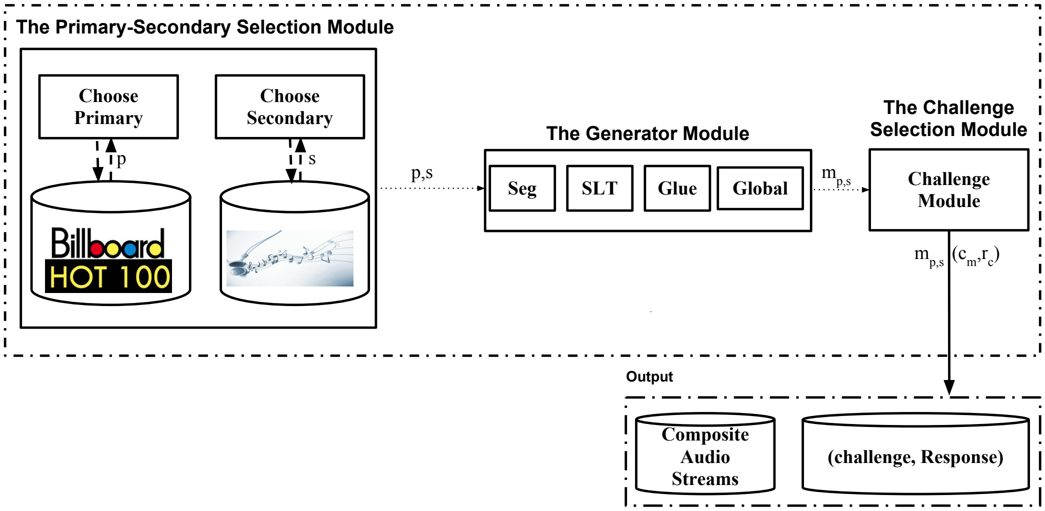


Figure 5.1: The mCaptcha system comprising of the primary-secondary selection module, the generator module, and the challenge selection module. The system outputs an mCaptcha composite music stream, a context challenge and the associated correct response. The generated composite stream can be heard as two by humans but is unlikely to be separated by automated programs.

5.2.1.3 Assumptions and Other Considerations

The new mCaptcha scheme assumes that the user is able to hear the generated music stream. That is, similar to the CAPTCHA model, the goal is to make the scheme accessible to as large a fraction of the population as possible [vABHL03].

The underlying demographic assumptions are based on current statistics available on digital music use. The rise and development of new technology that makes efficient access to music on a massive-scale possible, have created a new *digital reality* which will further be shaped by the "digital natives" [PG08, PGSB09]. In other words, our current technological reality not only has forced a change in past behavioural *norms* — such as means and duration of accessing or streaming music — but also it has nurtured youth who spend more time online and start using online resources at a much earlier age. The new generation of users may start using the internet as early as 10, and spend more than 3 hours online listening to music [JGB^{+}12].

Hence, it will be assumed that the majority of the target mCaptcha users will be between the ages of $10 - 45$. These users are without major hearing impairments, and can listen to the mCaptcha through their computers, mobile phones or other portable digital systems.

The chosen CAPTCHA challenge should be simple enough for human beings to solve. The CAPTCHA should be efficient to create even though they will be generated offline[5]. The music snippets, primary and secondary streams, are less than 15 seconds long, and the generated mCaptchas are manipulated remixes of these streams. The somewhat diminished aesthetic quality of the resulting composite and the short length of the building streams render potential copyright infringement objections moot. The particular choice of the challenge associated with an mCaptcha is an open design parameter, and will ensure that humans can solve the mCaptcha's challenge before the end of the allocated time t (efficiency), and computers will need more than t to correctly answer the challenge (security). Finally, in the security analysis of the scheme, I assume that the adversary does not

[5] In other words, mCaptchas can be generated and securely stored in a database for later access.

include humans posing as bots.

5.2.2 Security Considerations of mCaptchas

> "*There is no way to* `prove` *that a program cannot pass a test which a human can pass, since there is a program — the human brain — which passes the test.*"
>
> -Luis von Ahn et al. [vABHL03]

Security of this proposed system uses the CAPTCHA security framework (Section 5.1.2.2). Security is empirical, depends on the state-of-the-art algorithms, and evolves over time. Finally, if the mCaptcha scheme is *broken* by automated programs in the future, then a believed-to-be-difficult problem — at the intersection of music information retrieval (MIR) and computational auditory scene analysis (CASA) — can be efficiently solved. Consequently, a different set of music-based contextual challenges should be used.

In the context of mCaptchas, a requester is presented with a single audio stream which can be heard as two distinct music streams by humans, but which *appears* as a single inseparable unit of music to an automated adversary. In Section 5.2.2.1, security is justified in terms of increased size of the building alphabet (i.e., music streams instead of an alphanumerical one), random segmentation, combination of two valid but sufficiently distinct music streams, and a contextual challenge based on one of the music streams. Though a rigorous proof of security is impossible [vABHL03], evidence is provided in Section 5.4.1 through an investigation of resilience against attacks from a widely-used audio identification program. The ultimate test of mCaptchas, similar to other CAPTCHA protocols, is a large-scale adoption by web services, and the ultimate measure of its practical security is time.

5.2.2.1 Computational Difficulty of Adversarial Attacks

The adversary model used to break visual or audio CAPTCHAs relies on machine learning algorithms to detect speech components from a noisy background [TSHvA08, SC04]. The CAPTCHA is first broken into smaller components believed to contain an elementary unit, a letter or a number, and (speech) classifiers are applied to each segment. This adversarial model is not applicable to mCaptchas since speech classifiers are not sufficiently sophisticated to capture the information-rich structure and features in music.

The elementary unit of mCaptchas, instead of letters or digits, is music, and this choice highlights the system's complexity. By using music,"the composer has a large alphabet of possibilities. This alphabet is not a simple set of seven notes, but is the $350,000$ differentiable sounds in the full range of human hearing." [Coh62] Use of a more complex, information-latent medium — in place of 26 elements of construction — increases the amount of uncertainty [Wea49b] faced by automated, adaptive programs. Moreover, the number of songs available to be used for the primary and secondary streams are in the order of millions and this number continues to grow. For instance, the iTunes music stores — with 29% share of the music sales and available in 199 countries — contains 26 million songs with $15,000$ being songs downloaded per minute [Pha13], and the typical user has on average more than 8000 [BC09] songs in his iTunes library. This increased uncertainty and deluge of options implies that a state-of-the-art machine learning algorithm — absent the introduction of an ingenuous method to search through an exponentially-large space of possibilities efficiently — fails to correctly reconstruct the streams by randomly guessing.

The mCaptcha system is considered *broken* if a computational adversary can correctly answer challenges — contextual questions about primary music streams — corresponding to random generated mCaptchas in an allotted short period of time. The computational difficulty for breaking the mCaptcha relies on the following:

1. the computational difficulty of identifying the primary by separating the primary from the secondary music stream in the mCaptcha,

2. the computational difficulty of answering the contextual, primary-dependent challenge in an allotted, short period of time.

The challenge is chosen at random from a pool of possible contextual questions, and the correct response depends on a particular primary stream, chosen at random from a large collection of possible streams (millions of songs). These make guessing the response at random moot for the adversary.

I assume that the adversary has access to a large music collection, M^*, and that for each song there is associated *meta information*: $(s_i, t_{s_i}) \in M^*$. Such stored meta information may include name of the song, artist(s), album, year of release, information about genre and subjective tags (i.e., keywords such as emotional descriptors). Moreover, I assume that the adversary has access to any state-of-the-art algorithm that may be used to correctly answer the challenge, S^*. An mCaptcha instance is denoted by the composite music stream, m_{p_r,s_r}, a corresponding challenge, c_m, and a corresponding response, $\lambda(c_m)$. The mCaptcha's music stream, m_{p_r,s_r}, is built using randomly chosen primary (p_r) and secondary (s_r) songs as described in Section 5.2.1.2. Potential attacks to break an mCaptcha instance may be categorized into the following two general approaches:

- **Musical Stream Segregation:** Given m_{p_r,s_r}, the adversary applies source separation techniques to separate p_r from s_r: $I_{S^*}^{M^*}(m_{p_r,s_r}) = \tilde{p_r}$. It then extracts meta information stored for $\tilde{p_r}$ in M^* to answer c_m before the current mCaptcha instance expires.

- **Music Summarization:** The adversary extracts key local or global music attributes (e.g., spectral, rhythmic, or tonal information) from m, creates a music *signature* and compares this summarized description, $\tilde{m}$, against M^* to find p_r. Alternatively, a classifier (e.g., an state-of-the-art genre, artist or instrument classifier) may be applied to learn (partial or complete) information about the primary. The information learned by the adversary about m is denoted by: $I_{S^*}^{M^*}(\tilde{m})$. Finally, the adversary uses $I_{S^*}^{M^*}(\tilde{m})$ to correctly answer c_m before expiration.

The computational difficulty of the above two approaches can be summarized as:

- The problem of separating mixed auditory streams into their constituents is computationally difficult [Bre94]. Separating one stream from another — especially when the two are summed with random weights — is difficult because of the interference between their corresponding spectral components [Bre94].

- Despite current computational and algorithmic advances, exact automatic extraction of certain musical key attributes (e.g., time signature) from audio is computationally difficult [MEKR11].

- The problem of music source separation in this setting corresponds to *under-determined source separation* (USS) since there are more sources, (p_r, s_r), than audio mixtures

(m_{p_r,s_r}). USS, and more precisely music under-determined source separation, are particularly difficult [MEKR11]. Current approaches incorporate information about *timbre models*, *harmonicity* of the sources, *temporal continuity* and sparsity constraints [MEKR11]. Nonetheless, these remain computationally difficult.

- If the adversary successfully identifies p_r, answering certain contextual questions remains computationally difficult.

In Section 5.4.1, I present empirical results showing that a widely-used audio identification algorithm fails to identify the primary stream with high probability.

5.2.2.2 Large-Scale Deployment: Test of Time

From the initial deployment of reCAPTCHAs in May 2007 until July 2008 as a free service, over $40,000$ CAPTCHAs were incorporated into Websites and over 100 million CAPTCHAs were solved on a daily basis [vAMM$^+$08]. The ubiquity of music, the high number of culture-independent possibilities for primaries and secondaries, and the potential for improved Web accessibility for the visually impaired, are strong indicators for large-scale adoption of mCaptchas. Similar to CAPTCHAs, the true test of security will be resilience against attacks after a publicized massive-scale deployment.

5.2.3 Auditory Features Related to Usability

A key feature of the human auditory system is the ability to comprehend distinct components — originating from distinct sources — separately [Bre94, Dar97]. Auditory scene analysis (ASA), a term first coined by Bregman [Bre94], is devoted to the study of such

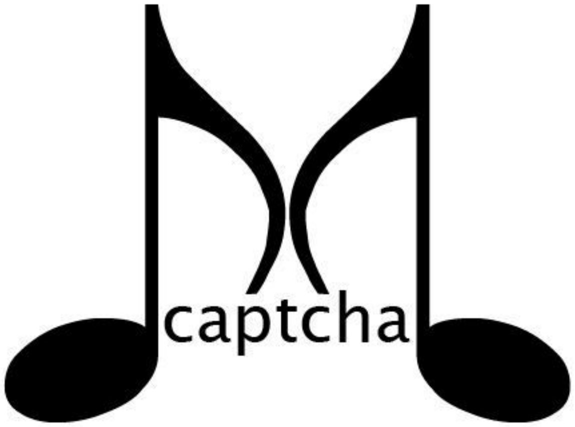


Figure 5.2: An example of a gestalt principle. Similarity and proximity of two distinct objects lead to the emergence of a new perceptual object. In other words, disparate shapes grouped together may be perceived as a single shape. For instance, the two distinct musical notes are perceived as an "M" here.

perceptual features. The ability of the auditory system to group and organize sound into meaningful components is similar to the visual grouping principles put forward by the school of *Gestalt psychology* [Bre94, Dar97]. The Gestalt grouping principles describe humans' unique ability to perceive complex patterns from simpler ones (*emergence*), generate information from limited portions of a stimulus (*reification*), ability to interchange between ambiguous perceptual stimulus (*multi-stability*), and classification of similar but manipulated objects together (*invariance*). For instance, in Figure 5.2 an "M" emerges from two distinct but similar objects.

Though an in-depth treatment of auditory scene analysis [BC71, BD73, BR75, DB76, Bre78, BP78, Bre94] is outside the scope of this work, I highlight the following principles relevant in the context of mCaptchas:

- The principle of *grouping*: auditory grouping depending on similarity and proximity,

  - **“Principle of harmonicity” [Bre94]:** regularities are used to detect multiple sources; grouped according to the most probable cause,
  - **“Envelope independence” [Bre94]:** asynchronies in intensity fluctuations lead to perception of multiple sources,
  - **“Streaming” [Bre94]:** Non-simultaneous sounds that are sufficiently similar — proximity in “acoustic distance” — are heard as one, and those with large acoustic distance as distinct streams.

- the principle of continuity, and
- the principle of past experience.

These principles account for why humans can hear the primary as a continuous stream, despite deleted chunks and the interruptions of the distractor (secondary stream); they also justify the detection of two distinct music streams from a mixture. Being able to hear two streams from a mixture, and detect a continuous primary stream are critical in recall and answering contextual questions about the primary musical stream.

To ensure that the primary and secondary streams are clearly heard as two distinct streams, music pieces with very different spectral signatures (“Frequency separation” [Bre94]) separated with sufficient temporal gaps (“Temporal separation” [Bre94]) are used. The temporal gaps — between blocks of the primary and secondary streams — and the choice of secondaries — with sufficiently distinct spectral signatures — are two heuristic parameters of design. The more different the primary is from the secondary — the notion of difference is based on information latent in the two streams (e.g., structural features such as rhythmic regularity) as well as contextual or culture (e.g., genre or musical style), the easier it is for an auditory segregation into two distinct streams [Bre94].

5.2.4 Applications of mCaptchas

My proposed scheme meets the requirements of CAPTCHA schemes outlined in Section 5.1.2. It relies on an information-potent ubiquitous commodity: digital music. The cornerstone novelty of the scheme is in its application of music and departure from security in the form of noise-based scrambled visual or audio signals. mCaptchas improve accessibility for users with visual impairments, remove dependence on a specific language, and allow for further customization to particular music libraries. They may be more pleasant because they avoid the use of noisy alphanumeric audio clips, and could even be entertaining to solve.The issue of improving user experience and web accessibility for the visually impaired is significant. The World Health Organization estimates that there are over 285 million individuals with visual impairments [PM12]. Article 21 of the United Nations Convention on the Rights of Persons with Disabilities - Freedom of expression and opinion, and access to information- mandates equal access to information for all [Ass06]. That is, users should not be deterred from access due to a disability. This novel design improves user accessibility by both using more pleasant building blocks instead of scrambled noise, and by allowing for a user-by-user customization. Music is unequivocally intertwined with numerous aspects of our daily lives [Ren12], and our innate ability to identify known pieces in matters of seconds [MB93, PG99] mark this design relevant for many applications.

The mCaptcha system, similar to its predecessors, is at the intersection of practical security and human computation. Possible applications for mCaptchas include, but are not limited to, access control, subjective user-generated music tags, and artist promotion. In the proceeding sections, I outline a few potential applications.

5.2.4.1 Access Control

The mCaptcha system can be used in all existing applications which currently use CAPTCHAs. Such applications include provision of access for online polls, online ticket brokers, registration process for web-based email service providers such as Gmail, online postings such as Wikipedia, blogs and forums and open government initiatives such as online government petitions [cap, vABL04, gov12, vABHL03].

5.2.4.2 Massive-scale Online Music Exposure

The CAPTCHA framework provides access to distributed human computation on a massive scale. For instance, the reCAPTCHA scheme was introduced as an enhancement of CAPTCHA [BvALH00, vABL02] to use the time and human processing power "wasted" on solving textual challenges for "good" [vAMM^{+}08]. This scheme replaced randomly selected alphanumeric letters, used in CAPTCHAs, with words from digitized books that optical character recognition (OCR) cannot accurately identify [vAMM^{+}08]. Von Ahn et al. note that within the course of fourteen months, over 100 million CAPTCHAs were solved on a daily basis [vAMM^{+}08].

There is a wealth of potential music streams to be used to generate mCaptchas. In 2012, there were over 1.5 billion digital singles downloaded [Fri12]. In May 2013, comScore estimated VEVO, a music channel on YouTube, to have close to 53 million unique users, watching over 600 million music videos [com13]. Other studies have reported over 3 hours per day of streaming [JGB+12] or listening to music on a portable device [BAG+12]. Lastly, a typical music library has thousands of songs [KBM12, BC09]. The mCaptcha system has two potential targets: advertisers and site owners. This provides the music industry the opportunity to direct the human user to upcoming concerts or promotional events through adaptive ads embedded in the mCaptcha. For independent artists with fan registration or polling services on their site, mCaptchas can be used to separate humans from bots while showcasing their own work as well as other artists who are in the same genre or music category.

5.2.4.3 Wisdom of the Crowds to Improve Music Recommendation

The future of music will be online. Cloud computing and online music streaming are no longer science fiction, and large companies such as Google, Apple and Amazon not only see that future but are actively working towards materializing this vision. In a near future, almost all music ever recorded will be made available online, and music recommendation engines will be a necessary tool. Because of the highly dimensional aspect of music, its subjectivity and reliance on human cognition, any feasible music recommendation engine will rely fully or partially on user-generated tags. The sheer amount of music available mandates the use of crowdsourcing for such a task. mCaptchas can be slightly modified to be further used as a massively-scalable tool to generate millions of user-generated tags. One particular approach, similar to reCaptcha [vAMM+08], is to ask a user to solve an mCaptcha and require him to give at least two tags that he associates with it.

5.2.5 Drawbacks

Similar to existing CAPTCHAs, both textual and audio, mCaptchas are susceptible to man-in-the-middle attacks. In these attacks an automated program or service provides incentives — financial or otherwise — for other humans to solve the challenges in place of the bots. Blum et al. refer to such attacks as “stealing cycles from humans” [vABL02].

5.3 An mCaptcha Prototype

A prototype of the mCaptcha system, described in Section 5.2.1.2, is presented here. Details of various modules (Figure 5.1) are given in Section 5.3.1. Security experiments are presented in Section 5.4.1, and finally usability is investigated in Section 5.5. I use an audio-identification algorithm to provide evidence for security and investigate the scheme’s usability on a crowdsourcing platform. The usability experiments provide an overview of the user-experience, response accuracy and efficiency for over 4000 mCaptchas (Section 5.5). The prototype (Figure 5.1) is implemented in Matlab, and uses a Matlab implementation of an audio identification algorithm by Ellis [Ell09] as an adversary; the security results for over 2000 (unique) mCaptchas are presented in Section 5.2.2.

5.3.1 Parameters of Design

The mCaptcha system outlined in Section 5.2.1.2 has access to two collections of music: the primary and the secondary. The primary music collection, P, contains songs selected from the Billboard Hot 100 [Fra91]. To ensure diversity and user ease of detection, at least ten popular songs were chosen from decades between 1960 and the present. The music snippets, each less than 15-seconds long, were downloaded from freely available music snippets on Wikipedia and Amazon's mp3 store. The secondary music collection, S, contains classical music snippets downloaded for free from Deutsche Grammophon, a German classical record label [dG]. The genres of the two collections are chosen to be very different (pop vs. classical) to ensure sufficient variation in musical attributes, such as structural regularity, between the primary and secondary streams.

An mCaptcha instance is a tuple: $(m_{p_r,s_r}, c_{m_{p_r,s_r}}, \lambda(c_{m_{p_r,s_r}}))$ where the music stream m_{p_r,s_r} is generated through the following sequence of phases:

- **Primary Selection:** p_r
  - Choose p_r randomly from the primary music collection, P.
- **Secondary Selection:** s_r
  - Choose s_r randomly from the secondary music collection, S.
- **Generator Module:**
  - Segmentation Phase (`Seg(.)`):
    - ∗ Segment the primary and secondary streams; segmentation lengths of the primary are chosen at random ranging from the 700 to 1000 ms,
    - ∗ Denote the primary and secondary segments by $P = [\langle P^1 \rangle \langle P^2 \rangle \ldots \langle P^{n_p} \rangle]$
    - ∗ The secondary segments are denoted by $S = [\langle S^1 \rangle \langle S^2 \rangle \ldots \langle S^{n_s} \rangle]$ with segments' lengths chosen at random:

      $|\langle S^i \rangle = r_i |\langle P^i \rangle|$, where r_i is chosen at random from (0.15,0.2).

- Stochastic Linear Transformation Phase (`SLT(.)`):
  - ∗ Two variations are implemented: discard blocks either at random or drop all even blocks. The result is: $\tilde{P} = \{\langle w_i^1 P_i \rangle\}_{i=1}^{n_p} = \{\langle \tilde{P}_i \rangle\}_{i=1}^{n_p}$ where,
    1. (Even Indices) $w_i^1 = 0$ for even indices,
    2. (Random Indices) $w_i^1 = 0$ for randomly chosen indices. However, consecutive blocks are not discarded to ensure that the resulting stream is aesthetically pleasing.
- Stochastic Combination Phase (`Glue(.)`): Choose weights uniformly at random from $(0.5 : 0.05 : 0.7)$, pad the secondary as needed, and compute a weighted average $\{(w_i^2 \tilde{P}_i + w_i^3 S^i)\}_{i=1}^{n_p}$, with $w_i^2 + w_i^3 = 1$

- **Challenge Generation Module:** The following two challenges are implemented:

  1. **Name That Tune Case:** name the artist, album or song,
     - In the first case, the user will be granted access so long as the name of the song or the album entered are approximately correct - allowing for small typos by the human users.
  2. **Comparison Case:** which sounds older?
     - the user will listen to two mCaptchas and select which one sounds older. The two mCaptchas created use building block music pieces that are at least three decades apart. Studies have shown that human beings are great at detecting the decade information about music piece even if the user is not familiar with the piece.

- **Verification of the User's Response:** The provided human response should match the meta-information corresponding to the mCaptcha stored in a secure database on the provider's servers.

Finally, to investigate whether visual aids further improve user-experience, the user is presented with a mosaic of various cover arts with the cover art corresponding to the primary magnified (Figure 5.3).

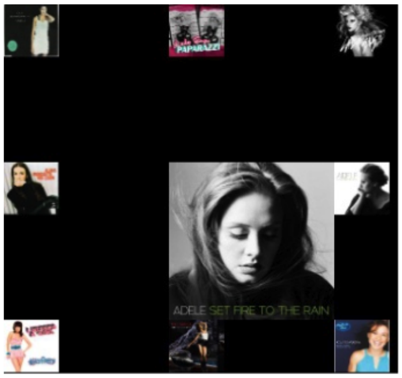


Figure 5.3: A possible visual aid. The mosaic of various cover arts includes a magnified cover art corresponding to the primary stream. This is to visually help the user rememer the name of the album, song or the artist.

5.4 A Security Evaluation of the Prototype

As discussed in Section 5.2.2, a rigorous *proof* of security is not possible for mCaptchas. The ultimate test of security, similar to existing CAPTCHAs, will be the test of time after a large-scale adoption. However, this section provides evidence for security by investigating the success of a widely-used audio identification program in detecting the primary stream.

I use Shazam, a commercial music identification service [WSI00], to evaluate the ease of detecting the primary or the secondary streams, given access to the mCaptcha and the underlying music collections (primary and secondary music collections).

Developed in the early 2000 [WSI00], Shazam's music identification system is able to correctly identify very short segments of music despite strong background noise and interference in less than 10 seconds [Wan03]. The great success of its mobile application [sha] in correctly and efficiently identifying songs from very short noisy or distorted samples recorded and transmitted from mobile phones, has turned it into a great financial success [Kin11] with a great following of over 300 million users [sha13].

This audio identification algorithm computes an *audio fingerprint* for a noisy, distorted short song sample and compares it to musical fingerprints of already stored millions of songs [Rei08]. More precisely, for each song a time-frequency map (spectrogram) is computed; Local peaks of the spectrogram are selected ("constellations"). From the constellations, focal points with clusters around them ("anchors") are chosen; For each frequency-time pair (f_j, t_j) in the neighbourhood around an anchor (f_a, t_a), $(f_a, f_j, t_a - t_j)$ are stored; These, along with the time offset from the beginning of the stream and track ID are stored as fingerprints The algorithm is robust against distortions and noise since a true peak in the original will most likely remain a peak after the stream has been filtered or distorted by adding noise [WSI00]. The collected peaks in the spectrogram are referred to as a constellation [Wan03]. From the constellation, anchors are chosen. These are focal points in the constellation with clusters around them. For each point (f_j, t_j) in the neighbourhood around anchor (f_a, t_a), $(f_a, f_j, t_a - t_j)$ are stored. These values along with the time offset from the beginning of the stream and track ID are stored as fingerprints and used to determine a match [Wan03].

Shazam's efficiency, accuracy and large scale adoption make it a great candidate for an state-of-the-art adversary in my security analysis of mCaptchas. I use an open-source implementation of Shazam by Ellis [Ell09]. I use this Matlab implementation — instead of Shazam's mobile application — in my analysis to ensure that the adversary has access to all primary and secondary songs, and that the failure to identify the primary (or the secondary) is not due to a lack of access to classical music. This particular audio identification algorithm is, henceforth, referred to as *the adversary*.

5.4.1 Security Results

The adversary used in the following tests forms pairs from of frequency peaks ("landmarks") in the audio; For each pair, frequency values along with time offsets in between these local maxima are stored and further quantized to approximately $20 - 50$ landmarks per second [Ell09]. The music collection, of size 393, consists of all primary and secondary streams in addition to a few audio streams not used in the generation of mCaptchas. These fingerprints are computed for a music collection containing all primary and secondary streams, and matched against those computed for mCaptchas. For an mCaptcha stream, m_{p_r,s_r}, the adversary returns a list of top 10 matches, $A(m_{p_r,s_r}) = \{a^i_{m_{p_r,s_r}}\}_{i=1}^{10}$. The following cases are significant in the security analysis

1. **No matches.** The mCaptcha is *secure*:

   - $\nexists i \in [1, 10], a^i_{m_{p_r,s_r}} = p_r$,
   - $\nexists j \in [1, 10], a^j_{m_{p_r,s_r}} = s_r$,

2. **Primary stream is found.**

    - $\exists i \in [1, 10], a^{i}_{m_{p_r,s_r}} = p_r$

        - If the primary is returned as the first choice (i.e., $a^{1}_{m_{p_r,s_r}} = p_r$), the mCaptcha is *broken*.

3. **Secondary stream is found.**

    - $\exists j \in [1, 10], a^{j}_{m_{p_r,s_r}} = s_r$

In addition to the security of generated mCaptchas, denoted by $M = \{m_{p_i,s_i}\}_{i=1}^{N}$ where N is the total number of mCaptchas generated, the influence of two design parameters on security are investigated: the length of the segmentation blocks and choice of which blocks are discarded (i.e. are multiplied by weights $w_i = 0$). A security score, s_m, is computed which depends on the likelihood that the adversary fails (cannot identify the primary and the secondary stream) and his partial successes; a (partial or complete) success implies a correct identification of the primary (analogously the secondary) stream in the adversary's list, $A(m_{p_r,s_r}) = \{a^{i}_{m_{p_r,s_r}}\}_{i=1}^{10}$. A higher penalty is assigned for matched primaries than secondaries and to finding matches higher up in the list. The following indicator variables (i.e., binary variables that taken on 1 only when a particular condition is satisfied) are used define a security score (Definition 5.1):

- $\forall j \in [1, N], \forall i \in [1, 10]$,

    - (**Primary match at index** i):

        - $I^{i}_{p_j} = 1$, iff $a^{i}_{m_{p_j,s_j}} = p_j$,

- (**Secondary match at index i**):
  - $I^i_{s_j} = 1$, iff $a^i_{m_{p_j,s_j}} = s_j$,
- (**No matches**):
  - $I_{m_{p_j,s_j}} = 1$, iff $\Sigma^{10}_{i=1} I^i_{p_j} + I^i_{s_j} = 0$.

Let the security score, s_M, for an mCaptcha collection, $M = \{m_{p_j,s_j}\}^N_{j=1}$, be defined as:

$$s_M = \Sigma^N_{j=1}(I_{m_{p_j,s_j}} - \Sigma^{10}_{i=1}(\frac{I^i_{p_j}}{i} + \frac{I^i_{s_j}}{i^2})) \tag{5.1}$$

An mCaptcha stream, $m_{p_r,s_r} \in M$, is constructed from randomly chosen, distinct primary and secondary music streams, p_r and s_r. These streams are subjected to stochastic segmentation (`Seg`), stochastic linear transformation (`SLT`), stochastic combination (`Glue`) and a global modification (Section 5.3.1). I include two intermediate collections of music streams, M_e and M_r, in the security analysis. These collections consist of streams before the application of any global non-linear transformations (final step in the mCaptcha prototype - Section 5.3.1). The following variations are considered:

- M_e **("Drop Even Blocks"):** set weights w_i in `Seg`$(.)$ to zero for even indices,
  - $\hat{M}_e$: Choose segmentation lengths at random from $(700 - 1000)$ ms.
  - $\bar{M}_e$: Choose segmentation lengths at random, and drop random blocks (but use longer gaps filled with the secondary.)
- M_r **("Drop Random Blocks"):** Choose indices i where $w_i = 0$ at random
  - $\hat{M}_r$: Choose segmentation lengths at random from $(700 - 1000)$ ms.

  - $\bar{M}_r$: Choose segmentation lengths at random, use longer gaps filled with the secondary (extend by factor e) and ensure that no two adjacent blocks are dropped. The length of each secondary block will be an e multiple of the length of its most immediate primary block neighbour. The extension factor, e, is chosen at random, from a fixed small set of possible values, for each stream (e.g., $e = 1.2$).

Table 5.1 summarizes the security results of mCaptchas tested. Here, the adversary is Shazam, an audio identification algorithm, and mCaptchas were constructed using 248 primary (popular songs taken from Billboard top 100 songs [Fra91]) and 107 secondary streams (Western classical music taken from Deutsche Grammaphone [dG]). Shazam was given access to both the primary and secondary collections, in addition to 37 other audio signals (e.g. birds chirping). Based on Table 5.1, I make the following observations:

- **Case 1 ($700 - 1000$):** Segmentation blocks in the primary are at least 700 and at most 1000 ms long.
  - Generated mCaptchas, M, are resilient against audio identification attacks: over 98% of the final mCaptcha streams, M, were undetectable against Shazam [Wan03, Ell09].
  - Dropping even blocks is harder to detect than random blocks (compare 27% to 48% rate of failure); presumably because in the random case, adjacent blocks may contain structural or harmonic information useful for audio identification,
    - $\hat{M}_r$ **vs.** $\bar{M}_r$**:** By ensuring that no two adjacent blocks in the primary are

  dropped at random, $\bar{M}_r$, the adversary can potentially detect latent information and security suffers. This is reflected in Shazam's increased success percentage of approximately 10%. In other words, randomness in choosing which blocks of the primary are silenced in $\bar{M}_r$ is biased, and adjacent blocks of the primary are likely to contain enough temporal-spectral information that Shazam can correctly identify the primary. In that case (p_i, s_j, p_{i+1}), the secondary block in between the two contiguous primary blocks will be treated as *noise* by the adversary.

- **Case 2 (**$1000 - 1200$**):** Segmentation lengths are increased to the $1000 - 1200$ ms range. Although in this case, the streams may sound slightly more pleasant, the likelihood of detection by the adversary also increases (5% instead of the previous 2%).
  - Similar to the previous case, dropping even blocks of the primary results in a slight disadvantage for Shazam (27% vs. 35%).
- In both cases, dropping random blocks makes detection more difficult for Shazam.

	Final mCaptcha Streams	Intermediate Streams			
		Drop Even Blocks		Drop Random Blocks	
	(M)	(M_e)		(M_r)	
Segmentation length: $700 - 1000$ **ms**	$\lvert M \rvert = 2232$	$\lvert \hat{M_e} \rvert = 744$	$\lvert \bar{M_e} \rvert = 744$	$\lvert \hat{M_r} \rvert = 744$	$\lvert \bar{M_r} \rvert = 744$
p_r and s_r are not found	2135(96%)	201(27%)	525(71%)	346(47%)	281(38%)
Segmentation length: $1000 - 1200$ **ms**	$\lvert M \rvert = 744$	$\lvert \hat{M_e} \rvert = 248$	$\lvert \bar{M_e} \rvert = 248$	$\lvert \hat{M_r} \rvert = 248$	$\lvert \bar{M_r} \rvert = 248$
p_r and s_r are not found	709(95%)	66(27%)	160(65%)	88(35%)	92(37%)

Table 5.1: Influence of mCaptcha design parameters on security. Generated mCaptchas, M, are the streams presented to a requester. "Intermediate streams" columns show the influence of different design parameters on the likelihood of the adversary's success.

Table 5.1 highlights all-or-nothing scenarios of security: either the mCaptcha is broken (i.e., the primary is found) or not. However, security compromises may be partial: the adversary may find the primary or the secondary as candidate streams in his return list of 10, $A(m_{p_r,s_r}) = \{a^i_{m_{p_r,s_r}}\}^{10}_{i=1}$. Although a partial match is not as significant as the previous scenario, I present evidence that the proposed scheme — in its final form (M) and its intermediate phases — remains resilient with high probability. Security scores are computed according to Equation 5.1, and Figures 5.4 and 5.5 show the likelihood of the adversary achieving such security scores. These scores and their associated probabilities are computed for results of Table 5.1. Figure 5.4 shows that the adversary fails identify the primary and the secondary with overwhelming probability, and that the probability it finds the primary or the secondary as its first response ($a^1_{m_{p_r,s_r}}$) is almost zero: $Pr[s_m = -1] \approx 0$.

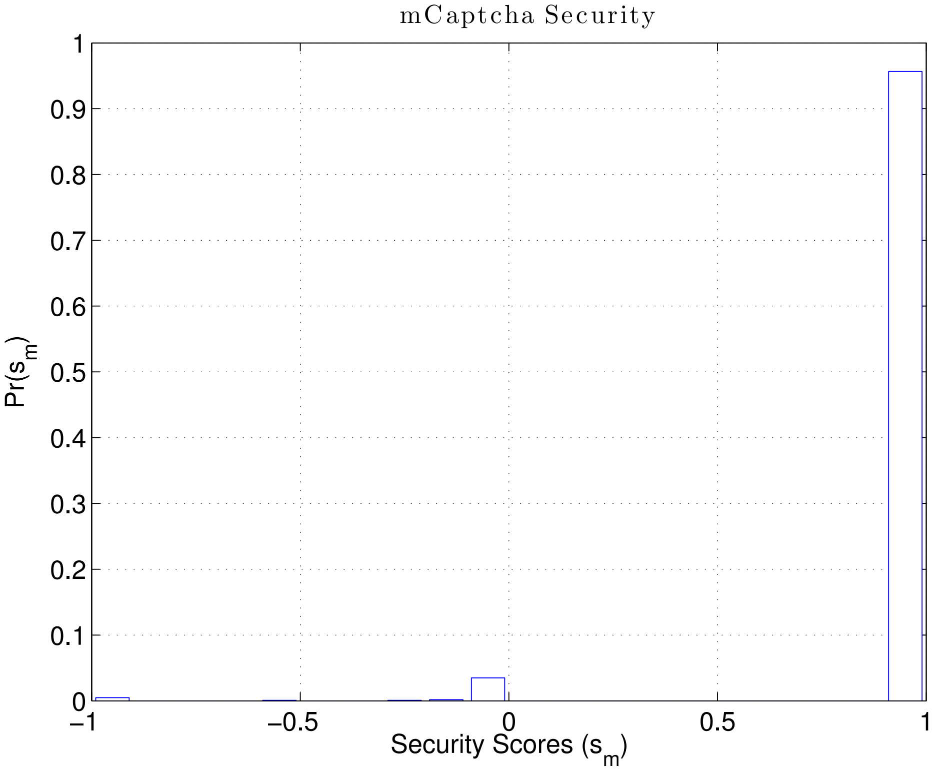
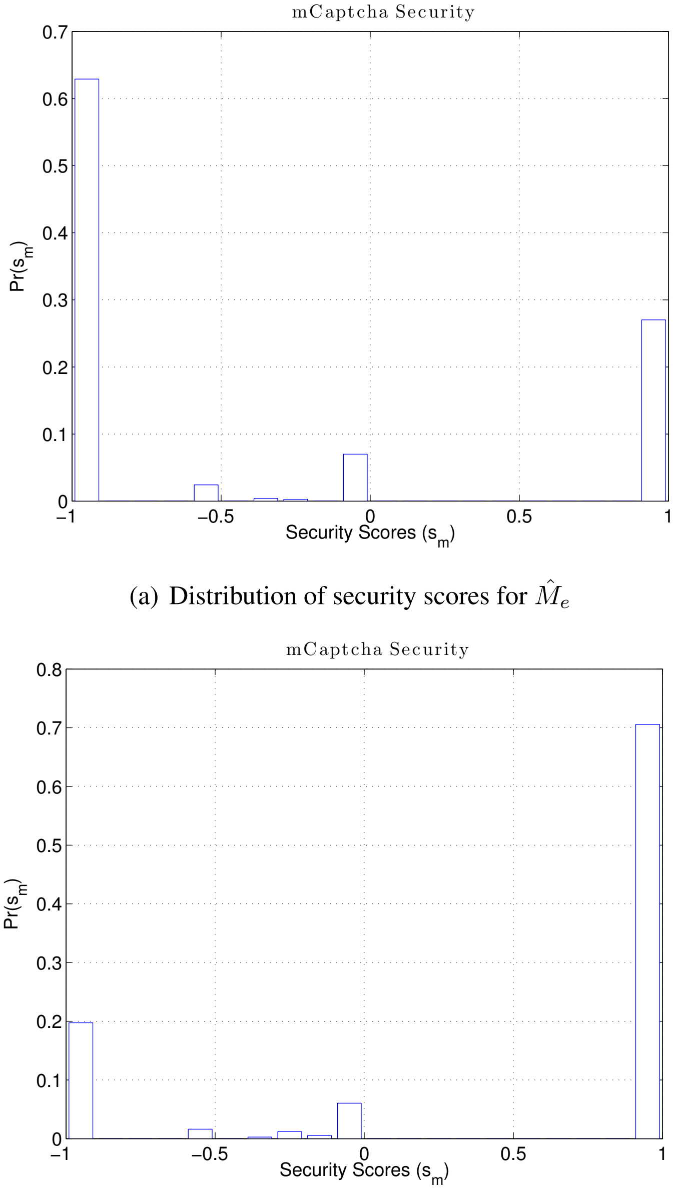


Figure 5.4: Distribution of security scores of final mCaptchas (M): A score of 1 corresponds to the primary and secondary not being found, and scores less than 0 correspond to partial matches of the primary or the secondary weighted by significance of the match (Equation 5.1).

Figure 5.4 also highlights that partial matches further down in the adversary's list of candidate stream, $\{a^i_{m_{pr},s_r}\}_i$, is very unlikely: $Pr[-1 < s_m < 0] < 0.1$. The following observations are of note about the security scores:



(a) Distribution of security scores for $\hat{M}_e$

(b) Distribution of security scores for $\bar{M}_e$

Figure 5.5: Distribution of security scores (intermediate case of M_e): A score of 1 corresponds to the primary and secondary not being found, and scores less than 0 correspond to partial matches of the primary or secondary weighted by significance of the match (Equation 5.1).

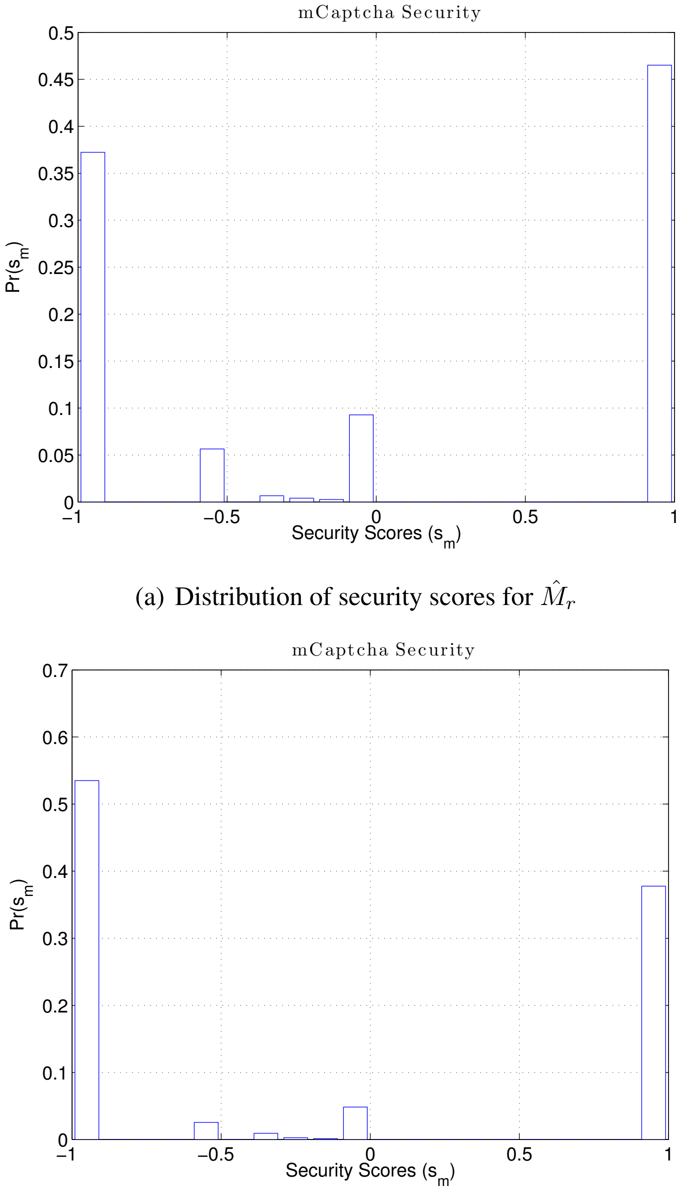
- **Security scores $\hat{M}_e$ vs. $\hat{M}_r$:** Although by dropping even blocks, the adversary less likely to correctly identifying the primary in M_r (i.e., the probability of a security score 1 — corresponding to no matches — is higher in $\hat{M}_r$ than in $\hat{M}_e$), it has a higher probability of correctly identifying the primary *or* the secondary (corresponding to security score $s_m = -1$ (equation 5.1)) in $\hat{M}_e$.

- **Security scores $\hat{M}_e$ vs. $\bar{M}_e$:** Figure 5.5(b) shows that by increasing the lengths of gaps in between primary segments, security scores improve: $Pr[s_{\hat{M}_e} = 1] > Pr[s_{\bar{M}_e} = 1]$.

- **Security scores $\hat{M}_r$ vs. $\bar{M}_r$:** By ensuring that no consecutive blocks of the primary are discarded, the number of non-matches for both the primary and the secondary decreases (Table 5.1). In other words, the adversary can correctly identify the primary. However, Figure 5.6 shows that there is a decrease in the number of partial matches for the primary or the secondary; there is a decrease in the likelihood of security scores $-1 < s_m < 0$.

5.4.2 Discussion of the Security Results

Similar to CAPTCHAs [vABHL03], proof of security in this context is impossible. Therefore, I do not attempt to give a formal proof of security but rather provide empirical evidence for the computational difficulty of attacks. The security results of Section 5.2.2 demonstrate the resilience of generated mCaptchas against a state-of-the-art adversary.



(a) Distribution of security scores for $\hat{M}_r$

(b) Distribution of security scores for $\bar{M}_r$

Figure 5.6: Distribution of security scores (intermediate case of M_r): A score of 1 corresponds to the primary and secondary not being found, and scores less than 0 correspond to partial matches of the primary or the secondary weighted by significance of the match (Equation 5.1).

Though the results are promising, the ultimate test of security is time[6], and requires a large-scale deployment of the scheme. In the following section, mCaptchas that passed the security test (i.e., those on which Shazam failed to detect both the primary and the secondary streams), are used in a series of large-scale user tests, wherein I investigate how easy it is for humans to solve mCaptchas accurately.

5.5 Usability Analysis of mCaptchas

The notion of usability measures the efficiency and accuracy of solving mCaptcha challenges by humans. I show that mCaptchas are user-friendly through large-scale experiments on a popular crowdsourcing online market, Amazon Mechanical Turk (AMT) [mTu]. This crowdsourcing platform is well suited for problems which are trivial for humans but computationally complex or expensive otherwise [KCS08, KK08, MBR10, Ipe10a, MS12, PCI10, Ipe10b]. Examples of problems solved on AMT include photo tagging, handwriting recognition and iterative text improvement [LCGM09, PCI10].

5.5.1 Mechanical Turk Parameters

In mCaptcha usability tests, human participants, `workers`, are assigned mCaptchas, `HITs`, and paid a small fee ($\$0.01 - \$0.05$) for the successful completion of each HIT , which stands for "Human Intelligence Task". In my experiments, the majority of workers were paid a penny, and a few were paid 5 cents to solve mCaptchas and answer a few

[6] I borrow the notion of "test of time" from cryptography where certain cryptographic primitives, such as practical encryption schemes, are considered to be *strong* if the state-of-the-art attacks by the best cryptographers continue to fail over time [FS03].

usability questions. Workers participating after June 2013 were paid 5 cents to encourage *only* new users to participate in the study in a short period of time. I examined the user experience in solving mCaptchas by measuring the duration needed to solve, and also by asking users to rank their experience, on a scale of 1 (very pleasant) - 4 (unpleasant). HITs were presented in a random order, and batches were uploaded at various times of a day to avoid targeting a particular worker demographic. Each HIT is solved by at least 3 workers, and each worker must solve the HIT without any interruptions. HITs expire after 5 minutes of inaction, and each batch of HITs is considered finished when all assignments have been completed or 10 days have passed.

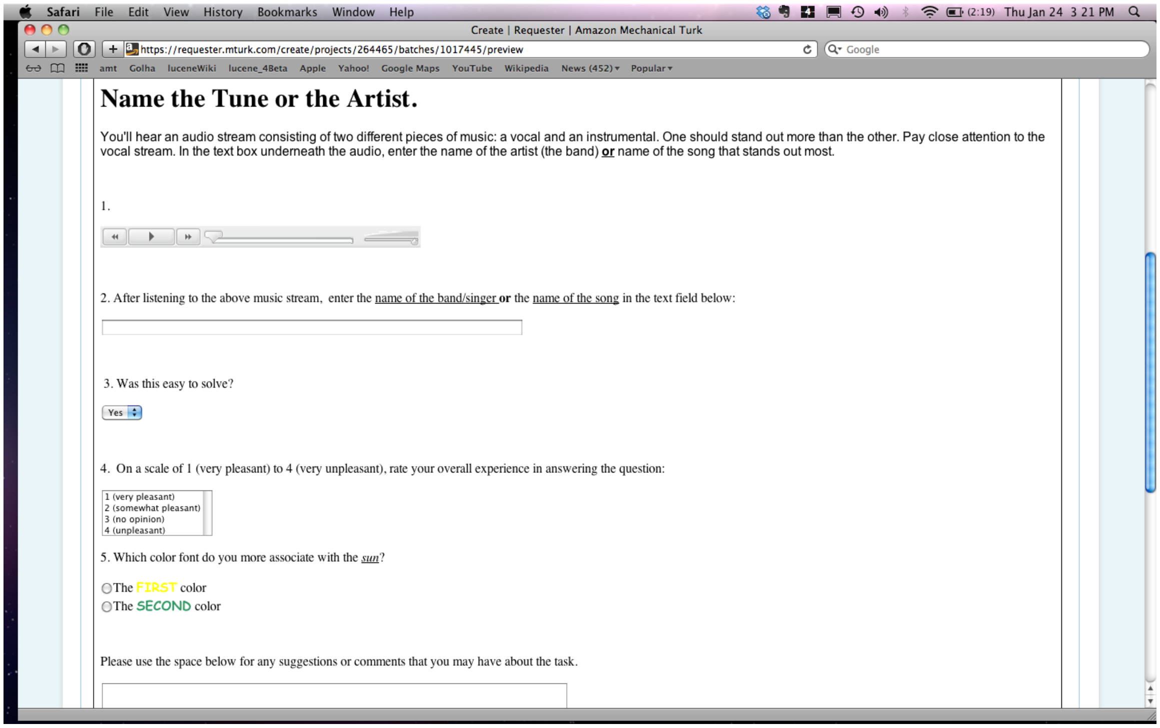
I conducted three variations on AMT [mTu]:

1. **Name That Tune without Visual Aid.** The user listens to a composite music stream and is asked to name the tune or the artist (Figure 5.7)

2. **Name That Tune with Visual Aid.** This case investigates if visual aids can improve efficiency or accuracy: a mosaic is shown to the user, containing the current primary's cover art along with others chosen at random (Figure 5.8), and

3. **Compare the Decades.** In this final category, the user is presented with two composite streams and asked to identify the composite stream whose primary stream belongs to a more recent decade. For instance, the first mCaptcha contains an Adele song ($1^{\mathtt{st}}$ decade of the $21^{\mathtt{st}}$ century) whereas the second contains a piece by Aretha Franklin (recorded 50 years earlier). All instances tested were at least 30 years apart. A total of 3104 mCaptchas (156 unique mCaptchas) were solved by 290 unique workers. The most active worker solved a total of 165 mCaptchas (Figure 5.9).

In addition, workers were asked to rate their experience and to indicate whether or not they found the task easy. For quality control, a simple test is put in place to distinguish humans from bots on AMT and ensures that users are not answering the mCaptcha challenges at random. Workers who failed these tests were not paid, their responses were rejected and the mCaptchas were returned to the pool of available HITs.


Safari File Edit View History Bookmarks Window Help
(2:19) Thu Jan 24 3 21 PM
Create | Requester | Amazon Mechanical Turk
https://requester.mturk.com/create/projects/264465/batches/1017445/preview
Google
amt Golha luceneWiki lucene_4Beta Apple Yahoo! Google Maps YouTube Wikipedia News (452) Popular

Name the Tune or the Artist.

You'll hear an audio stream consisting of two different pieces of music: a vocal and an instrumental. One should stand out more than the other. Pay close attention to the vocal stream. In the text box underneath the audio, enter the name of the artist (the band) **or** name of the song that stands out most.

1.

2. After listening to the above music stream, enter the name of the band/singer **or** the name of the song in the text field below:

3. Was this easy to solve?

Yes

4. On a scale of 1 (very pleasant) to 4 (very unpleasant), rate your overall experience in answering the question:

1 (very pleasant)
2 (somewhat pleasant)
3 (no opinion)
4 (unpleasant)

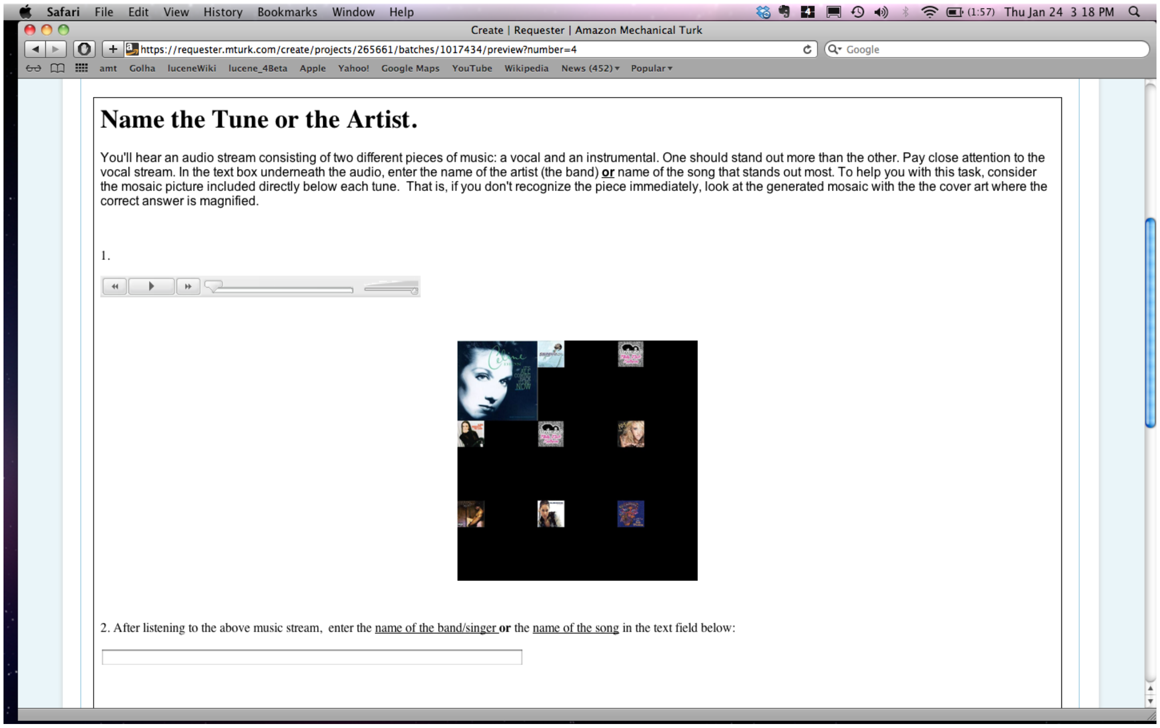
5. Which color font do you more associate with the *sun*?

○ The FIRST color
○ The SECOND color

Please use the space below for any suggestions or comments that you may have about the task.

Figure 5.7: mCaptchas: Name that Tune Case. No visual aids are provided.

In the “Name that Tune without Visual Aid” case, users listen to a composite music stream and answer the challenge: name the tune or the artist — whichever they found to be easier.


Safari File Edit View History Bookmarks Window Help
(1:57) Thu Jan 24 3 18 PM
Create | Requester | Amazon Mechanical Turk
https://requester.mturk.com/create/projects/265661/batches/1017434/preview?number=4
Google
amt Golha luceneWiki lucene_4Beta Apple Yahoo! Google Maps YouTube Wikipedia News (452) Popular

Name the Tune or the Artist.

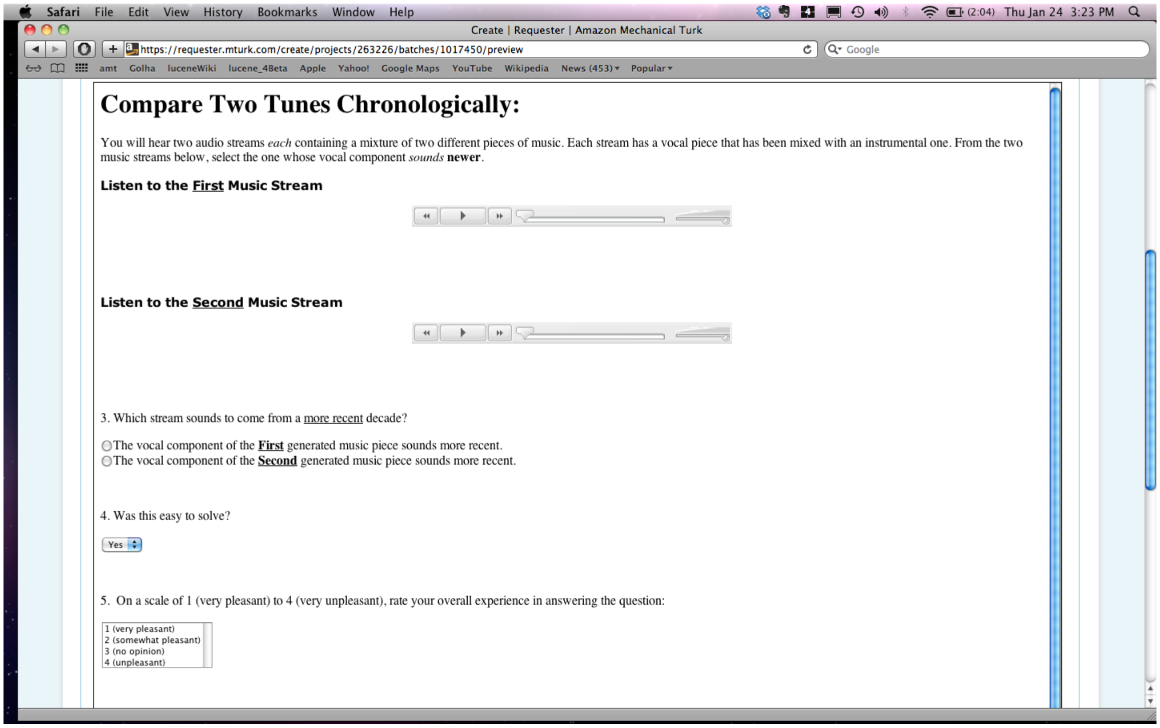
You'll hear an audio stream consisting of two different pieces of music: a vocal and an instrumental. One should stand out more than the other. Pay close attention to the vocal stream. In the text box underneath the audio, enter the name of the artist (the band) **or** name of the song that stands out most. To help you with this task, consider the mosaic picture included directly below each tune. That is, if you don't recognize the piece immediately, look at the generated mosaic with the the cover art where the correct answer is magnified.

1.

2. After listening to the above music stream, enter the name of the band/singer **or** the name of the song in the text field below:

Figure 5.8: mCaptchas: Name that Tune Case. A mosaic of various album cover arts of different sizes are presented. The cover art corresponding to the correct answer is the one most prominently visible.

The "Name that Tune with Visual Aid" case was considered to investigate if visual aids are necessary, and whether or not they can sufficiently help users not familiar with the presented stream. The final category of experiments is the "Comparison Case." In this scenario, the user was presented two composite streams, and is asked to identify the mCaptcha whose primary stream belongs to a more recent decade. For instance, the first mCaptcha consists of an Adele song (1st decade of the 21st century) whereas the second contains a piece by Aretha Franklin (recorded 50 years earlier). All instances tested were



Compare Two Tunes Chronologically:

You will hear two audio streams *each* containing a mixture of two different pieces of music. Each stream has a vocal piece that has been mixed with an instrumental one. From the two music streams below, select the one whose vocal component *sounds* **newer**.

Listen to the First Music Stream

Listen to the Second Music Stream

3. Which stream sounds to come from a more recent decade?

○The vocal component of the **First** generated music piece sounds more recent.
○The vocal component of the **Second** generated music piece sounds more recent.

4. Was this easy to solve?

Yes

5. On a scale of 1 (very pleasant) to 4 (very unpleasant), rate your overall experience in answering the question:

1 (very pleasant)
2 (somewhat pleasant)
3 (no opinion)
4 (unpleasant)

Figure 5.9: mCaptchas: Comparison case. The user is presented with two mCaptchas and asked to mark the one whose primary stream sounds more recent.

at least 30 years apart.

To protect against automated responses during my tests, I placed in a few simple questions for the users to answer. If a user did not pass this simple test, his response was rejected. The rejected mCaptchas were placed back in the pool of possible mCaptchas to be solved. Table 5.5.1 lists questions along, with possible choices. A worker's *approval rating*, measured in percentages, and marks his rate of success passing these rudimentary tests.

Question	**Possible Choices**
Which can be the next logical item in the following list: Ferrari, Porsche, and Lamborghini?	a BMW a turtle
In *reality*, which of the following can fly?	a Ferrari an eagle
Which one is closer to the colour of the ocean?	1. (`red text`) 2. (`blue text`)
In *reality*, which can you possibly drive?	a magic carpet a Ferrari
In *reality*, who can you have lunch with?	Mozart your best friend
In *reality*, who can you have dinner with?	your best friend Einstein

Table 5.2: Different basic tests put in place to ensure that user was not randomly answering the questions.

5.5.2 Description of the Users in this Usability Analysis

My experiments ran through January - July 2013 and contracted

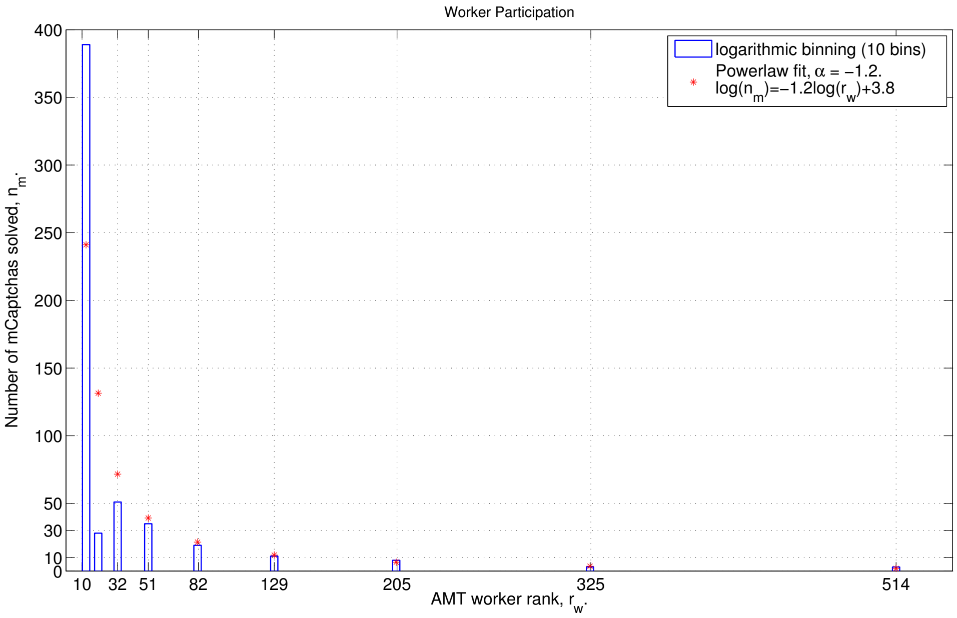


Figure 5.10: The power law in AMT worker participation (Name that Tune and Comparison cases). The Workers are sorted in a decreasing order of the number of mCaptchas they solved, forming their ranks, r_w. The number of mCaptchas solved, n_m, follows a power law: That is, $n_m \propto r_w^\alpha$ where $\alpha = -1.2$.

- 218 workers for the "name that tune with no aid" case,
- 55 workers for the "name that tune with aid" case, and
- 290 workers for the "comparison" case.

The same worker may have participated in different categories, and each mCaptcha was solved by at least 3 workers. A total of 490 unique workers participated. The worker participation distribution follows a power law: many users solve few mCaptchas and very few many (Figure 5.10).

Even though some mCaptchas were redundant, they were assigned at random to various workers. Moreover, the majority of workers were not prevented from solving mCaptchas of different types; however, new workers[7] were given a higher financial incentive (5 cents instead of 1) in the final days of the experiment. Finally, each worker was required to answer (1) the usability — ranking their overall experience — and (2) quality questions — demonstrating that they are indeed humans by solving a simple question (e.g., Table 5.5.1) — before submission.

5.5.3 Efficiency and Accuracy Results

In this context, efficiency is the amount of time (measured in seconds) that a Mechanical Turk worker spent on solving an mCaptcha, and accuracy is computed as fraction of mCaptchas correctly solved, over all responses and for all workers. It should be noted that workers were asked miscellaneous questions, to rank their overall experience, and that the durations discussed below include duration needed to answer such questions as well. In other words, the efficiency measures discussed here are conservative, and will, in reality, be smaller.

Tables 5.3, 5.4, and 5.5 include the efficiency results of various mCaptchas tested in my usability analysis. These are presented as evidence for feasibility of using mCaptchas.

[7] The workers were “non-master”; the master vs. non-master qualification granted by Amazon has no direct implications on the quality of work done. Amazon charges more for master workers.

$\|M\| = \|\{m_i\}_i\| = 4211$	No Aid
	Durations (sec)
$E[d_i]$	29.48
$\sigma(d_i)$	26.10
$E[d_i - \|m_i\|]$	18.94
$\sigma(d_i - \|m_i\|)$	23.29

Table 5.3: Efficiency Results - “Name the Tune with No Visual Aid” Case. A total of 4211 mCaptchas were solved.

The efficiency overview of the scheme can be further refined by subtracting the amount of time each user spent on listening to a mCaptcha stream. After deducting the duration of each mCaptcha music mixture, m_i, the resulting mean efficiencies ($\mu[\{d_i\}_{m_i \in M}] \pm \sigma(\{d_i\}_{m_i \in M})$) are: (1) (Name that Tune (without aid)) 18.94 seconds ($\sigma = 23.29$), (2) (Name that Tune (with aid)) 26.78 seconds ($\sigma = 42.51$), and (3) (Comparison) 26.09 seconds ($\sigma = 28.10$).

$\lvert M\rvert = \lvert\{m_i\}_i\rvert = 513$	With Aid
	Durations (sec)
$E[d_i]$	38.85
$\sigma(d_i)$	45.78
$E[d_i - \lvert m_i\rvert]$	26.78
$\sigma(d_i - \lvert m_i\rvert)$	42.51

Table 5.4: Efficiency Results - "Name the Tune with Visual Aid" Case. A total of 513 mCaptchas were solved.

In the Name that Tune (with aid) case (Table 5.4), the increase in mean duration and variation can be attributed to users taking time to comprehend the visual hint. The variation in user response time in the Comparison Case (Table 5.5) is comparable to that of the Name that Tune (no aid) case(Table 5.3). Its challenge does not require a user to remember the names of artists or songs, and requires no particular information to be read. As such, such mCaptchas are accessible options for individuals with mobility and visual impairments.

A total of 290 workers solved 3104 comparison mCaptchas. The correct answer correctly identified which of two mCaptchas presented sounded to belong to a more recent decade.

$\|M\| = \|\{m_i\}_i\| = 3104$	Comparison
	Durations (sec)
$E[d_i]$	39.01
$\sigma(d_i)$	34.60
$E[d_i - \|m_i\|]$	26.09
$\sigma(d_i - \|m_i\|)$	28.10

Table 5.5: Efficiency Results - Comparison Case. A total of 3104 mCaptchas were solved.

In the Name that Tune Case, from the 138 unique mCaptchas tested in the Name that Tune case, Roxette's "Joy Ride" (1991) was the hardest mCaptcha — solved by 5 distinct workers and requiring $\mu \pm \sigma = 82.40 \pm 80.13$ seconds — while Cher's "Believe" (1998) was the easiest — solved by 9 workers who took 16.33 ± 6.2 seconds. In addition, Gnarls Barkley's "Crazy (2006) was the song with the highest duration of 291 seconds; while, Aerosmith's "Don't Wanna Miss Anything" (1998) was the most efficient, with a duration of only 6 seconds.

	No Aid	With Aid	Comparison
$\|M\| = \|\{m_i\}_{i=1}^n\| = n$	4211	513	3104
Accuracy	0.96	0.97	0.73

Table 5.6: Mechanical Turk accuracy results

The accuracy results presented in Table 5.6 demonstrate that humans are able to solve mCaptchas correctly. For instance, Name that Tune (no aid) mCaptchas yielded over 96%

accuracy. In this case, 3114 (74%) of the user responses were exact matches to the correct response; 908 (22%) were partially matches (i.e., had few typos but were deemed to be correct) and only 189 (4%) were rejected. The Comparison mCaptchas demonstrated much lower accuracy, which may be caused by workers on AMT being more inclined to select one of the two tunes randomly to minimize their time spent on the task.

5.5.4 Aesthetic Quality Indicators

My investigation showed that the majority of users found mCaptchas to be somewhat or very pleasant, 84% in the name that Tune case and 81% in the comparison case (Table 5.7).

	No Aid	With Aid	Comparison
$\|M\|$	4211	513	3104
	$\%$ Participants	$\%$ Participants	$\%$ Participants
Very Pleasant	54	32	42
Somewhat Pleasant	30	42	39
No Opinion	9	17	14
Unpleasant	8	8	4

Table 5.7: User experience as rated by AMT workers reported in this usability analysis.

User experience and enjoyment are key factors that affect efficiency and accuracy of the system. For instance, users who did not find it easy to solve mCaptchas took longer to solve the posed challenge as well: That is, the mean and standard deviations of the time needed by users can be categorized as $\mu_{\texttt{Easy}} \pm \sigma = 25.7 \pm 18.4$ seconds versus $\mu_{\texttt{Not Easy}} \pm \sigma = 46.5 \pm 39.8$ seconds in the Name that Tune case. However, this influence is not drastic, and the mean durations further categorized by user experience are very close:

- $\mu_{\texttt{Very Pleasant}} \pm \sigma = 25.4 \pm 17.3$ seconds,
- $\mu_{\texttt{Somewhat Pleasant}} \pm \sigma = 28.0 \pm 22.9$ seconds,
- $\mu_{\texttt{No Opinion}} \pm \sigma = 31.2 \pm 28.9$ seconds, and
- $\mu_{\texttt{Unpleasant}} \pm \sigma = 30.1 \pm 25.0$ seconds.

The usability tests outlined in Section 5.5 not only show that mCaptchas can be accurately and efficiently solved, but also that users could find solving them pleasant. This is reflected in both direct user feedback (Table 5.7). In an attempt to demonstrate that users can enjoy solving mCaptchas, I introduce the addictive factor. In my usability analysis, it became apparent that various users took little time to start a new mCaptcha after the completion of another. More precisely, I denote the addictive factor, $A_w(.)$ for worker w, to be the expected time that passes once w submits mCaptcha $m^w_{t_i}$ and until w's acceptance of the next mCaptcha, $m^w_{t_{i+1}}$. That is, for each worker w solving a collection of n mCaptchas — $M^w = \{m^w_{t_1}, \ldots, m^w_{t_i}, m^w_{t_{i+1}}, \ldots, m^w_{t_n}\}$ — the addictive factor is measured as: $A_w(M^w) = \frac{1}{n-1}\Sigma_{i=2}^{n-1}(t_{i+1} - t_i)$. In this usability study, the addictive factors are less than a minute long on average (Table 5.8).

In this usability study, the mCaptchas were presented to the workers at random and the users were paid very little (between $1 - 5$ cents per mCaptcha). In this case, the addictive factor may be interpreted as how eager users, which were included in my study, were

	Total Number of Workers	Mean (sec)	Standard Deviation (sec)
Comparison Case	179	49	195
Name the Tune (no visual aid)	159	45	121
Name the Tune (with visual aid)	54	17	30

Table 5.8: Expected time passed between a worker's submission and his acceptance of the next mCaptcha.

to solve mCaptchas. That is, in this case, the addictive factor outlined in Table 5.8 may indicate that users enjoyed solving mCaptchas.

5.5.5 Discussion of the Usability Results

The results of the large-scale user studies of Section 5.5.3 are presented as evidence that mCaptchas are easy to solve. Three mCaptcha variations were tested: name that tune (with visual aid), name that tune (without visual aid), and compare decades. In total, 7828 mCaptchas were solved by 490 different individuals. The typical user spent:

- **Name that tune (without aid):** 18.94 ($\sigma = 23.29$) sec,
- **Name that tune (with aid):** 26.78 ($\sigma = 42.51$) sec,
- **Comparison:** 26.09 ($\sigma = 28.10$) sec.

In the case of standard textual CAPTCHAs, user studies based on two different sets of 1000 randomly chosen users, show that, on average, a textual CAPTCHA can be solved in 13.51 ± 6.37 seconds and a reCaptcha for 13.06 ± 7.67 seconds [vAMM^{+}08]. In comparison, a set of 1000 randomly chosen users from the Mechanical Turk usability study (Section 5.5), can solve mCaptchas in 18.57 seconds (Table 5.9).

	Name that Tune (No Aid)	Comparison
$\lvert M\rvert = \lvert\{m_i\}_{i=1}^n\rvert = n$	1000	1000
Accuracy	0.95	0.72
$E[d_i]$	29.19	38.18
$\sigma(d_i)$	26.79	34.64
$E[d_i - \lvert m_i\rvert]$	18.57	25.99
$\sigma(d_i - \lvert m_i\rvert)$	24.51	29.40

Table 5.9: Accuracy and efficiency results for a sample of 1000 randomly chosen users.

It should be noted that these durations include the time needed for a user to answer whether the mCaptchas was "easy" to answer, to rank their overall experience, and to answer a rudimentary question demonstrating that the mCaptcha is not solved by a bot (Figures 5.9 and 5.8). Answering these usability questions increases the *effective* time needed to solve an mCaptcha by at least a few seconds.

The complete usability tests, Table 5.6, demonstrate that a typical user correctly solved "Name that tune" mCaptchas with 96% — analogously, 97% in the "Name that tune with aid" case — accuracy. These are comparable to the accuracy results of 96.1% reported for reCaptcha [vAMM$^+$08]. However, accuracy in this context is not dependent on a user's familiarity with a particular language (e.g., English) or the ability to correctly spell [vAMM$^+$08]; use of music enables a global integration of particular globally known songs (e.g., mCaptchas using Adele or John Lennon's songs). Finally, allowing for approximate user answers — due to imperfect human memory recall — allows humans to avoid solving multiple challenges unnecessarily because of mistypes.

5.6 Conclusion and Future Work

In this chapter, I developed a novel music-based computational Turing test. The mCaptcha scheme was introduced as an efficient, secure, accurate scheme to tell humans apart from computer bots. The new scheme improves on CAPTCHAs by enhancing accessibility for individuals with visual impairments or those who may not be comfortable with English. Moreover, it exhibits improved security and usability, and will not be susceptible to current attacks against (audio) CAPTCHAs. Unlike any other existing access control measures, it takes advantage of complex structure of music and music cognition: particular characteristics of human comprehension of music, high dimensionality of music. It can be used to improve significant problems in the field of music information retrieval such as music keyword (tag) generation and music recommendation.

I introduced the building blocks of the scheme (Section 5.2.1.2) and described one possible implementation in detail (Section 5.3). I provided evidence for its improved security and usability attributed to the incorporation of complex, information-rich building blocks which may easily be processed and comprehended by humans.

Security of mCaptchas, similar to that of CAPTCHAs, is empirical and cannot be rigorously proven [vABL02, vABL04]. However, in this context, security relies not on the obfuscation by noise (e.g., audio and traditional (text-based) CAPTCHAs), but instead on the computational difficulty of correctly distinguishing between two valid musical streams and answering a contextual challenge about one.

I provided preliminary evidence for the security of the scheme through tests against a state-of-the-art audio identification algorithm (Section 5.4.1). Lastly, mCaptchas were shown to be easy for humans through tests on the Amazon Mechanical Turk, an online crowdsourcing market place. These usability results indicated that humans are able to solve mCaptchas efficiently and accurately.

5.6.1 Contributions

To the best of my knowledge, there exist no similar music-based computational Turing test schemes. The objectives of this new scheme were to improve accessibility, usability and security. The scheme is significant as it

- incorporates music in place of letters and numbers in the questions (the challenge) posed to a requester,
- uses a music-based contextual challenge,
- uses an alternative notion of security without relying on noise embeddings,
- is a great candidate for large-scale deployment:
  - can be efficiently generated,
  - it is customizable,
  - it is not restricted to a particular language (e.g., English).
- exhibits (potentially) improved security,
- exhibits (potentially) improved usability,

- improves accessibility for individuals with visual impairments and those who may not be very comfortable with English,
- can be used to further generate contextual, subjective knowledge about music on a massive scale.

5.6.2 Future Directions

A massive-scale adoption of mCaptchas can further solidify the suggested improved security and accessibility based on the preliminary results of Sections 5.4 and 5.5.

5.6.2.1 Personalized mCaptchas

Future implementations can be further customized to a particular user's musical taste (e.g., using primaries from a different genre), past preferences, or online music library. The more users are exposed to a particular type of music, the stronger the cognitive prediction models that enable speedy recognition [Til08]. Consequently, customizations based on individual (cloud) music libraries will both improve efficiency and accuracy as the user will almost certainly be familiar with his/her own music collection. Lastly, instead of selecting songs from a pre-determined list of *popular* songs, primary streams can be selected to be songs that are determined to be "popular" either through crowdsourcing (e.g., selected as the most popular from a few candidate song collections by Mechanical Turk workers) or through automated analysis of social informatics (e.g., user-generated information from Twitter, Facebook, etc.).

5.6.2.2 Subjective Tag Generation

> *"Before the Internet, coordinating more than* 100, 000 *people, let alone paying them, was essentially impossible. . . . [Now,] we've gotten* 750 *million people to help us digitize human knowledge."*
>
> - Luis von Ahn [TED11]

With a slight modification, mCaptchas can be leveraged to collect user-generated subjective music keywords (e.g., mood descriptors such as sad, happy, and angry) for each primary music stream. Also, an implementation of mCaptchas as a two-player game — wherein the objective of the game is to solve the mCaptchas in the shortest amount of time — can be envisioned. In such a game, the genre of the next mCaptcha can be chosen at random or by the winner of the previous round, scores against other players can be accumulated in real-time or stored and updated over a period of time, as more games are played by the two particular players. In such a game, players can also be rewarded bonus points by entering keywords related to the primary which either match those of the other player, or those already stored from previous games — higher bonus points can be assigned to "less popular" keywords.

6

Conclusions and Remaining Work

"*If a composer could say what he had to say in words he would not bother trying to say it in music.*"

- Gustav Mahler [Mah96]

Music is a rich form of communication and emotional expression [Mil00, Cro01]. Its ubiquity manifests itself with cultural specificity [Bro91, BBN95] and its information-rich structure and features render its analyses non-trivial. While scientific research on musical analysis is by no means sparse, most works in the field of Musical Information Retrieval (MIR) have focused on applying computational tools to extract features from various aspects of this information-bearing stream, and have leveraged those features to distinguish between different musical types, characteristics or forms. Informally, the two common threads in many such problems are the application of computational tools to simplify the informational structure of music to a set of representative features (*lifting the veil of information deluge*), and the computational use of the resulting discriminants to categorize (*information-based segregation into disjoint categories*). This thesis takes a different viewpoint. It targets the computational analyses of features extracted from musical rhythmic structure in (1) music attributed to Western European or American composers (i.e., the Western collection analyzed in Chapter 3) and (2) non-Western music (i.e., African, Chinese, Persian and Turkish music pieces available in my analyses, presented in Chapter 4. Lastly, in Chapter 5, the complexity and high information content of music is used to improve a practical computational application. The seemingly disparate theoretical and practical focuses of this dissertation intentionally target some of the intrinsic dualities in music (e.g., veil of information deluge compared to the ease of comprehending music by humans): That is, the temporal scale-free (1/f) analysis and the mCaptcha scheme both wrestle with a complexity-simplicity duality. To rigorously investigate the duality in this information-bearing medium, I developed or adopted computational tools that

- investigated the structural patterns in music to better understand predictability and surprise features in symbolic music samples (simplicity),
- investigated and extracted structural information in the form of self-similarity from music (simplicity),
- demonstrated that this non-trivial, latent information is simple and efficiently computable (simplicity),
- analyzed the temporal structure for music available from various parts of the World and different time eras (manifestation simplicity despite diversity and complexity),
- presented evidence that such simple descriptors convey sufficient information to distinguish between various sub-categorizations of music (complexity),
- studied complex musical features and problems in music to determine a computational gap between human musical capabilities and those of automated programs

(computational complexity versus simplicity),

The work presented in this thesis draws from research in music cognition and machine learning, and its novel contributions can be broadly summarized as follows:

1. it extended the body of work on power law exponents in musical rhythm to compositions from Africa, China, Iran and Turkey By doing so, it shows that the emergence of temporal scale-free patterns in music are not limited to particular geographical, cultural or historical time periods,

2. it used the scale-free temporal exponents in musical rhythm to classify compositions by their "composer" labels and by a broad geographical binary dichotomy as Western vs. non-Western,

3. it leveraged the complexity of music and particular features of music cognition to create new music-based computational Turing tests. This novel integration of music enhances the security and accessibility of the existing computational framework.

The resulting analyses and all corresponding discussions, in the context of Chapters 3 and 4, were limited to the symbolic music samples available. In other words, music collections analyzed in my dissertation were limited to pieces available in online music libraries such as KernScores[Sap05]). The contributions of each chapter are further reviewed below.

6.1 Temporal Scale-Free Signatures

6.1.1 Western Music (Chapter 3)

In this chapter, I studied scale-free features of musical rhythm for the Western music collection , $R_{\texttt{Western}}$ described in Section 3.1. These features were used as a concise temporal representation of the structural repetitions in music from the 16^{th} to the 20^{th} century. The Western music collection ,$R_{\texttt{Western}}$, categorized by the name of composers, consisted of compositions by 24 composers (Figure 3.1) with corresponding styles ranging from Baroque to Musical theatre. The collection comprised 1165 symbolic music pieces; each represented concisely as a progression of notated durations in time (music time series). In this rhythmic representation, silences were treated equally as significant as notes: their durations were also stored in the music time series. Lastly, in the presence of various instruments (voices), durations for various voices were merged, while preserving temporal order. The Detrended Fluctuation Analysis (DFA) [PBH$^+$94] was applied to compute a temporal power law exponent, α, for each music time series. These exponents, along with other temporal similarity features — encapsulating the information latent in musical structure — were used as concise identifiers (signatures). Variations and similarities in values of α were highlighted for different composers, various compositions by a particular composer, and distinct compositions grouped according to some common characteristic of their corresponding composers (e.g., musical era). Lastly, I presented detailed analyses for the temporal scale-free exponents of compositions by four composers with geographically diverse backgrounds and distinct musical styles (Table 3.3).

More precisely, my work directly built on existing research by Levitin et al. [LCM12], and it made the following main contributions in Chapter 3:

- This work expanded the power law characterization of rhythm in [LCM12] by
  - **Incorporating an alternative toolkit for computer-aided musicology.** I used a new, object-oriented computational toolkit, `music21` [CA10]. This alternative approach led for the first time in such work to the inclusion of MIDI files [SF97] in the symbolic music collection analyzed.
  - **Expanding the scope of Western analysis.** Though the Western collection used in this analysis overlapped with that of [LCM12] (both use the Kern-Scores library [Sap05]), inclusions of composers such as Gershwin, from the $20^{\mathtt{th}}$ century with distinct compositional styles, uniquely expanded the scope of analyses and set it apart from [LCM12].
  - **including different temporal fractal exponents.** My novel analyses focused on α, computed using the DFA algorithm [PHSG95] for its robust behaviour in case of non-stationary processes; however other temporal power law exponents from structural repetitions in musical rhythm were also computed. More precisely, whereas Levitin et al., focused on power law exponents of spectra computed for rhythms of Western compositions [LCM12], my analyses computed spectral power law exponents, DFA exponents (α) [PHSG95], and Hurst exponents (H) [Hur51] for each music time series. The spectral exponents, $\beta_{\mathtt{interpolate}}$, were implemented with identical parameters as those of [LCM12], and were consistent with results presented therein. Two other spectral exponents, with slightly different parameters, were also computed. In the

  time-domain, two power law correlation exponents were computed: α and H. These have been used to detect long-range correlations in time series, and their applications as composer signatures are novel.

  - **Providing a granular analysis of temporal scale-free exponents in rhythm.** This work provided a focused study of variations and similarities of such exponents on a composer-by-composer basis.

- This work showed that the mean values of α for Western classical compositions , analyzed in this thesis, were in the fractal range, $\alpha \in (0, 1)$; these results are consistent with previous research on power laws in the spectra of Western rhythms [LCM12] but extend that work in novel directions, scope, and depth.

- Moreover, the granular composer analysis presented here highlighted variations in α for various compositions by a particular composer. This variation was further studied in correspondence with a composer's biography, whenever possible.

- The influences of a composer's musical era, other coeval composers, and his geographical origin on the fractal exponents were tested. This analysis demonstrated that capturing the structural preferences of composers using the previous factors falls short to fully capture the richness latent in distinct compositional styles of different composers. Although these factors may contribute to similarities in temporal scale-free exponents, these factors may not be sufficient conditions to accurately predict or explain similarities between two repertories. Nonetheless, the following categorizations based on era and origin were considered. The following similarities and disparities were noted, *only in as so far as* relating to the music samples analyzed in this dissertation:

  - Composers with similar professional timelines (Table 3.1):
    - Similar DFA exponents:
      - Giovannelli ($1583 - 1624$) - Monteverdi ($1582 - 1643$).
    - Disparate DFA exponents:
      - J. S. Bach ($1703 - 1749$) - Vivaldi ($1703 - 1739$)

- Composers with the same country of birth and hence native language (Table 3.2):
  - Similar DFA exponents: Beethoven - Brahms,
  - Disparate DFA exponents: J. S. Bach - Beethoven.

- Similar DFA exponents were observed for composers whose rivalry or whose influence on one another is well documented (e.g., Beethoven and Brahms)

- The following cases of similarities in values of α for particular composers were noted:
  - Distinct compositions, in a particular year, having similar scale-free structural characteristics.
  - Repetitions of structural patterns in a particular composer's distinct compositions separated by an extended period of time
    - Grieg's "op03-4" (1863) and "op46-4" both have $\alpha = 0.98$, and are separated by almost a decade (Figure 3.12)

- Since all composers in this Western collection demonstrated some degree of variation in their fractal exponents, I further focused on four Western classical composers. More precisely, compositions by George Gershwin ($1898 - 1937$), Edvard Grieg ($1843 - 1907$) from the Romantic era, Wolfgang Amadeus Mozart ($1756 - 1791$) from the Classical era, and Domenico Scarlatti ($1685 - 1757$) from the Galant musical style were analyzed. The four composers are geographically diverse; they hail from the USA, Norway, Austria and Italy respectively. I presented a detailed analysis of α for the compositions attributed to these composers , that were available in $R_{\texttt{Western}}$, and highlighted compositions that had anomalous exponents (with corresponding biographical notes when available).

  - variation in values of α where observed even for compositions in the same year (e.g.,"Hight Hat" and "How Long Has This Been Going On") composed by Gershwin in 1927 (Table 3.5))

- Finally, classification of various compositions based on temporal features were presented here for the first time. The classification accuracy demonstrated the high information-content value of such exponents, and provided further evidence for their application as discriminative descriptors:

  - The classification accuracy results demonstrated that fractal exponents carry enough information to be used for decade, composer and genre classification. The following binary classification cases were analyzed:

    - **Major composers:** Bach, Beethoven, Haydn, Frescobaldi, Joplin, Mozart, Scarlatti, Schubert, Sousa and Vivaldi,

    - ∗ **Disparate composers:** The classification analyses here considered composers with different Musical eras, whom hailed from different countries and lived in distinct time periods (e.g., Corelli vs. Frescobaldi)
    - ∗ **Stylistically affiliated composers:** Here, I considered composers with documented evidence of influence (e.g., Joplin vs. Sousa or Mozart vs. Bach)
    - ∗ **Composers with similar time era:** This classification further focused on composers in this Western collection whom not only lived in approximately identical time periods, but that also were active professionally in similar periods (e.g., Bach vs. Vivaldi)
    - ∗ **Composers with the same first language** Classification analysis included only composers that hailed from the same country (e.g., Bach vs. Beethoven), and finally
    - ∗ **Compositions grouped by century:** Three categories were classified:
        - · early to mid $17^{\mathtt{th}}$ century,
        - · mid to late $18^{\mathtt{th}}$ century,
        - · early to mid $20^{\mathtt{th}}$ century.
  - – To determine the influence and significance of fractal exponents, in comparison with other predictability exponents such as surprise indices and entropy included in the feature sets (Table 2.4.4), classifications of composers from the case studies (Gershwin, Grieg, Mozart and Scarlatti) were repeated by using all exponents, only predictability exponents only fractal exponents (spectral, α and H), or only α. My analysis shows that the temporal scale-free exponents

provide complimentary information in classifications in that classification accuracy increases by using all exponents together. The extent of this improvement depends on the structural style of each composer. For instance, scale-free temporal features were more influential in classifications involving Mozart's collection in my study (e.g., Mozart vs. Scarlatti), and less so in other classifications (e.g., Gershwin vs. Scarlatti).

6.1.2 Non-Western Music (Chapter 4)

Chapter 4 studied power law features in the rhythmic structure of music originating from Africa, China, Iran and Turkey. The collection of symbolic music samples available for these regions was referred to as the non-Western collection, $R_{\texttt{non-Western}}$, in my dissertation for simplicity. The contributions of this chapter include:

- The novel extension of temporal scale-free correlation exponents in rhythm to music origination from Africa, China, Iran and Turkey,
- Demonstrating a commonality between the rhythmic structure in georgraphically-diverse music using temporal scale-free exponents
  - in the context of the analyzed music samples, this analysis showed that temporal correlations in the structure of music were not limited to a particular geographical region or culture.
- Demonstrating a successful application of mathematical analyses and techniques typically developed in the context of Western music to the music of other cultures (Question posed in [TKSW07]),

- Presenting highly-accurate classification results as support for the suitability of these temporal fractal descriptors to automatically distinguish between the Western and non-Western dichotomies, in the context of the music collections discussed in this dissertation.

6.2 mCaptcha: A Computational Use of Music's Complexity

Chapter 5 presented a novel integration of music in a ubiquitous application used as a security measure against automated programs. Music-based CAPTCHAs, mCaptchas, were introduced as a scheme wherein music's complexity and latent rich information are not hindering obstacles, but rather key contributing factors to improve security. The proposed scheme is the first of its kind to incorporate music and use unique features of the human musical cognition in a computational Turing test context. I motivated the need for such a new scheme and highlighted the general guidelines for creating efficient music-based schemes that accurately distinguishes between humans and automated programs. The architecture of the system and two prototypes were discussed. Empirical evidence for the following features were presented:

- **Security: Computational difficulty**
  - security in this context, unlike existing schemes, is not based on noise-embeddings, and relies on the computational difficulty of
    - distinguishing between two valid musical streams, and

    - accurately and efficiently answering a contextual question about one of the streams.

  - empirical security tests were performed against a widely-used audio identification algorithm,
  - these security tests demonstrated the resilience of the mCaptchas against identification:
    - a quantifiable security score, s_m, was introduced,
    - this score marks the failure of the adversary as 1, and penalizes this maximal score depending on which stream (primary or secondary) the adversary detects,
    - this score was computed for all mCaptchas generated for the security tests in addition to streams resulting from intermediate phases of the mCaptcha generation process,
    - security analysis of the intermediary streams highlighted the strength and necessity of various design components

- **Accessibility: Ease of use by humans:** A crowdsourcing platform was used to determine how easy it is for humans to solve mCaptchas correctly. This phase, included only mCaptchas that were demonstrated to be secure from the security analysis:
  - Usability results were tested for three variations of mCaptchas: (1)"Name that Tune" (no visual aid), (2)"Name that Tune" (with visual aid), and (3)"Comparison Case".

  - Human usability tests used crowdsourcing:
    - Amazon Mechanical Turk online market was used,
    - Over 4000 mCaptchas were solved by 500 "workers"
    - the usability results of Section 5.5 demonstrated that users are able to accurately and efficiently solve mCaptchas:
      - A typical user spent $18.94, 26.78$, and 26.09 seconds respectively for the three usability cases considered.
      - Moreover, it was shown that human efficiency in the context of mCaptchas is comparable to that of traditional, text-based CAPTCHAs (Section 5.5.5).
  - The large-scale access to human testers provided by the Mecshanical Turk online market was used to show that most humans found the task pleasurable (and were likely to become "addicted" to solving mCaptchas; the addictive factor for each user was computed as the mean time interval, measured in seconds, between the completion of an mCaptcha and the acceptance of the next).

- the scheme was designed with the following objectives:
  - to improve Web accessibility (e.g., for individuals with visual impairments),
  - remove any language barriers in solving such human vs. bot tests,
  - improve security, and
  - improve the overall user-experience in solving mCaptchas.

- Finally, the scheme was demonstrated to be efficient by presenting an actual prototype.

6.3 Outlook

This section briefly outlines future possible extensions of this dissertation. An immediate extension of this analysis is to apply the temporal power law correlation analyses to a larger collection, preferably one including composers or regions that were not available in machine-readable format at the time of this study. The ELVIS project — Electronic Locator of Vertical Interval Successions — with over 5000 pieces from the $14^{\mathtt{th}}$ to the $19^{\mathtt{th}}$ century is one such suitable candidate [elv]. The temporal analysis of repetitions in rhythm, can be applied to compositions with ambiguous attributions. For instance, fractal signatures may be used to shed some light on the veracity of compositions attributed to Josquin des Prez, a late $15^{\mathtt{th}}$ century Franco-Flemish composer [Eld89, She00].

Future implementations of mCaptchas can be further customized to a particular user's taste (e.g., using primaries from a different genre), past preferences, or online music library. Moreover, large-scale social informatics (e.g., information available on Facebook or Twitter about what is "cool" or popular) may be used instead of the pre-determined list of primary streams. Lastly, generates streams could be adapted for a particular music genre related music site such as Deutsche Grammophon [dG] or orchestral groups [VSO], or they can be customized for promoting works of particular artists.

With a slight modification, mCaptchas can be leveraged to collect user-generated subjective music keywords (e.g. mood descriptors such as sad, happy, angry) for each primary music stream. Also, an implementation of mCaptchas as a two-player game — wherein the objective is to solve the mCaptchas in the shortest amount of time — can be envisioned. In such a game, the genre of the next mCaptcha can be chosen at random or by the winner, scores against other players are accumulated in real-time or stored and updated over a period of time as more games are played by the two particular players. In this game, players can also be awarded bonus points by entering keywords related to the primary which either match those of the other player, or have already been stored — higher bonus points can be assigned to "less popular" keywords.

The fractal exponents computed in Chapters 3 and 4 provided a rich, granular characterization of music, which can be applied to mCaptchas: to determine which combinations of primary-secondary streams have a higher chance of remaining safe, and on the other hand, to be used to enhance audio identification algorithms by shortening the amount of time needed to search for a particular song. Namely, instead of choosing the secondary stream at random, one could choose a secondary stream with an appropriate fractal characteristic. Initially, a temporal self-similarity signature can be computed for each primary and secondary. This is proceeded by a classification and sensitivity analysis to determine if primaries and secondaries with similar fractal exponents are more likely to form secure pairs. Since similarities in fractal signatures indicate repetitions in the temporal structure, it would be more computationally difficult for an adversary to distinguish between two streams with similar fractal exponents. It should also be noted that fractal exponents may improve the state-of-the-art algorithms trying break mCaptchas. For instance, future implementations of Shazam, or similar audio identification algorithms, may use the discriminative power of these scale-free exponents can be leveraged to narrow down the audio search space and be beneficial to audio identification algorithms.

Classification results presented in Chapters 3 and 4 fall under the category of content-based classification. An enhancement would be to build classifiers which use both temporal scale-free exponents and subjective tags generated by millions of users who are solving mCaptchas. The fusion of these two information-rich techniques may be critical for future World music recommendation and search engines.

tocchapterNon-Western Binary Classification Tables

A

Detailed Non-Western Binary Classification Results

A.1 African - Chinese Music

Table A.1: Binary Classification of African and Chinese music. Logistic regression is used with 10 fold cross validation.

$\lvert R_{\texttt{African}}\rvert = 21$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Chinese}}\rvert = 69$	Classified (%)	Absolute Error	F-Measure	African	China	
Fractal Exponents, Π	70	0.32	0.69	6	15	African
				12	57	China
Π	70	0.33	0.69	5	16	African
				11	58	China
Fractal Exponents	72.2	0.32	0.67	2	19	African
				6	63	China
α	76.7	0.36	0.67	0	4	African
				0	69	China

Table A.2: Binary Classification of African and Chinese music. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

$\lvert R_{\texttt{African}}\rvert = 21$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Chinese}}\rvert = 69$	Classified (%)	Absolute Error	F-Measure	African	China	
Fractal Exponents, Π	67.8	0.33	0.68	6	15	African
				14	55	China
Π	65.6	0.36	0.66	6	15	African
				16	53	China
Fractal Exponents	73.3	0.45	0.65	0	21	African
				3	66	China
α	76.7	0.36	0.67	0	21	African
				0	69	China

A.2 African - Persian Music

Table A.3: Binary Classification of African and Persian music. Logistic regression is used with 10 fold cross validation.

$\lvert R_{\texttt{African}}\rvert = 21$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Persian}}\rvert = 47$	Classified	Absolute Error	F-Measure	African	Persian	
Fractal Exponents, Π	58.8	0.45	0.60	9	12	African
				16	31	Persian
Π	60.3	0.46	0.59	5	16	African
				11	36	Persian
Fractal Exponents	60.3	0.41	0.58	4	17	African
				10	37	Persian
α	69.1	0.42	0.57	0	21	African
				0	47	Persian

Table A.4: Binary Classification of African and Persian music. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

$\lvert R_{\texttt{African}}\rvert = 21$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Persian}}\rvert = 47$	Classified	Absolute Error	F-Measure	African	Persian	
Fractal Exponents, Π	72.1	0.3	0.72	11	10	African
				9	38	Persian
Π	61.8	0.43	0.58	3	18	African
				8	39	Persian
Fractal Exponents	66.2	0.44	0.57	1	20	African
				3	44	Persian
α	69.1	0.43	0.57	0	21	African
				0	47	Persian

A.3 African - Turkish Music

Table A.5: Binary Classification of African and Turkish music. Logistic regression is used with 10 fold cross validation.

$\vert R_{\texttt{African}}\vert = 21$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\vert R_{\texttt{Turkish}}\vert = 38$	Classified	Absolute Error	F-Measure	African	Turkish	
Fractal Exponents, Π	74.6	0.26	0.75	16	5	African
				10	28	Turkish
Π	69.5	0.30	0.70	15	6	African
				12	26	Turkish
Fractal Exponents	71.2	0.36	0.7	10	11	African
				6	32	Turkish
α	67.8	0.44	0.62	4	17	African
				2	36	Turkish

Table A.6: Binary Classification of African and Turkish music. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

$\lvert R_{\texttt{African}}\rvert = 21$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Turkish}}\rvert = 38$	Classified (%)	Absolute Error	F-Measure	African	Turkish	
Fractal Exponents, Π	67.8	0.32	0.67	9	12	African
				7	31	Turkish
Π	69.5	0.30	0.7	14	7	African
				11	27	Turkish
Fractal Exponents	62.7	0.42	0.61	7	14	African
				8	30	Turkish
α	64.4	0.45	0.53	1	20	African
				1	37	Turkish

A.4 Chinese - Persian Music

Table A.7: Binary Classification of Chinese and Persian music. Logistic regression is used with 10 fold cross validation.

$\lvert R_{\texttt{China}} \rvert = 107$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Persian}} \rvert = 47$	Classified	Absolute Error	F-Measure	Chinese	Persian	
Fractal Exponents, Π	77.9	0.29	0.77	95	12	Chinese
				22	25	Persian
Π	74.7	0.33	0.73	97	10	Chinese
				29	18	Persian
Fractal Exponents	72.0	0.38	0.68	100	7	Chinese
				36	11	Persian
α	69.5	0.42	0.58	106	1	Chinese
				46	1	Persian

Table A.8: Binary Classification of Chinese and Persian music. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

$\lvert R_{\texttt{China}}\rvert = 107$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Iran}}\rvert = 47$	Classified (%)	Absolute Error	F-Measure	Chinese	Persian	
Fractal Exponents, Π	66.2	0.34	0.66	82	25	Chinese
				27	20	Persian
Π	76	0.3	0.74	98	9	Chinese
				28	19	Persian
Fractal Exponents	68.2	0.4	0.62	98	9	Chinese
				40	7	Persian
α	69.5	0.42	0.58	107	0	Chinese
				47	0	Persian

A.5 Chinese - Turkish Music

Table A.9: Binary Classification of Chinese and Turkish music. Logistic regression is used with 10 fold cross validation.

$\|R_{\texttt{China}}\| = 107$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\|R_{\texttt{Turkey}}\| = 193$	Classified $(\%)$	F-Measure		Chinese	Turkish	
Fractal Exponents, Π	91	0.11	0.91	94	13	Chinese
				14	179	Turkish
Π	89	0.14	0.89	94	13	Chinese
				20	173	Turkish
Fractal Exponents	73.3	0.33	0.73	58	49	Chinese
				31	162	Turkish
α	65	0.45	0.54	5	102	Chinese
				3	190	Turkish

Table A.10: Binary Classification of Chinese and Turkish music. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

$\lvert R_{\texttt{China}}\rvert = 107$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Turkey}}\rvert = 193$	Classified(%)	Absolute Error	F-Measure	Chinese	Turkish	
Fractal Exponents, Π	88.3	0.14	0.88	88	19	Chinese
				16	177	Turkish
Π	85.3	0.16	0.88	83	24	Chinese
				20	173	Turkish
Fractal Exponents	70	0.36	0.7	57	50	Chinese
				40	153	Turkish
α	67	0.43	0.65	37	70	Chinese
				29	164	Turkish

A.6 Persian - Turkish Music

This observation holds for both logistic regression and decision trees.

Table A.11: Binary Classification of Persian and Turkish music. Logistic regression is used with 10 fold cross validation.

$\|R_{\texttt{Iran}}\| = 47$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\|R_{\texttt{Turkey}}\| = 81$	Classified (%)	Absolute Error	F-Measure	Persian	Turkish	
Fractal Exponents, Π	82.8	0.18	0.83	33	14	Persian
				8	73	Turkish
Π	79	0.24	0.79	31	16	Persian
				11	70	Turkish
Fractal Exponents	78.1	0.29	0.78	29	18	Persian
				10	71	Turkish
α	63.3	0.47	0.49	0	47	Persian
				0	87	Turkish

Table A.12: Binary Classification of Persian and Turkish music. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

$\lvert R_{\texttt{Persian}}\rvert = 47$	Correctly	Mean	Weighted Average	Confusion Matrix		
$\lvert R_{\texttt{Turkish}}\rvert = 81$	Classified	Absolute Error	F-Measure	Persian	Turkish	
Fractal Exponents, Π	85.6	0.18	0.83	34	13	Persian
				8	73	Turkish
Π	85.9	0.15	0.86	33	14	Persian
				4	77	Turkish
Fractal Exponents	74.2	0.28	0.74	27	20	Persian
				13	68	Turkish
α	63.3	0.46	0.49	0	47	Persian
				0	81	Turkish

tocchapterNon-Western Binary Classification Tables

B

Detailed Western Binary Classification Results

B.1 Major Western Composers

> “*You can’t have Bach, Mozart and Beethoven as your favorite composers. They simply define what music is!*”
>
> Michael Tilson Thomas [Wika]

This section presents results on binary classification of Bach, Beethoven, Haydn, Frescobaldi, Joplin, Mozart, Scarlatti, Schubert, Sousa, and Vivaldi.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Bach}}\rvert = 241, \lvert R_{\texttt{Beethoven}}\rvert = 173$				Bach	Beethoven	
Bach - Beethoven	96.6	0.04	0.97	232 5	9 168	Bach Beethoven
$\lvert R_{\texttt{Bach}}\rvert = 339, \lvert R_{\texttt{Haydn}}\rvert = 241$				Bach	Haydn	
Bach - Haydn	95.7	0.05	0.96	326 12	13 229	Bach Haydn
$\lvert R_{\texttt{Beethoven}}\rvert = 173, \lvert R_{\texttt{Haydn}}\rvert = 241$				Beethoven	Haydn	
Beethoven - Haydn	69.3	0.33	0.69	108 62	65 179	Beethoven Haydn

Table B.1: Binary Classification - Bach, Beethoven and Haydn. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly	Mean	Weighted Average	Confusion		
	Classified (%)	Absolute Error	F-Measure	Matrix		
$\lvert R_{\texttt{Bach}}\rvert = 241, \lvert R_{\texttt{Beethoven}}\rvert = 173$				Bach	Beethoven	
Bach - Beethoven	95.4	0.05	0.95	233	8	Bach
				11	162	Beethoven
$\lvert R_{\texttt{Bach}}\rvert = 339, \lvert R_{\texttt{Haydn}}\rvert = 241$				Bach	Haydn	
Bach - Haydn	96.6	0.05	0.97	326	13	Bach
				7	234	Haydn
$\lvert R_{\texttt{Beethoven}}\rvert = 173, \lvert R_{\texttt{Haydn}}\rvert = 241$				Beethoven	Haydn	
Beethoven - Haydn	72.2	0.33	0.72	110	63	Beethoven
				52	189	Haydn

Table B.2: Binary Classification - Bach, Beethoven and Haydn. Logistic regression, with 10 fold cross validation, is used.

	Correctly	Mean	Weighted Average	Confusion		
	Classified (%)	Absolute Error	F-Measure	Matrix		
$\lvert R_{\texttt{Corelli}}\rvert = 24, \lvert R_{\texttt{Joplin}}\rvert = 46$				Corelli	Joplin	
Corelli - Joplin (Logistic Regression)	90	0.10	0.9	22	2	Corelli
				5	41	Joplin
$\lvert R_{\texttt{Corelli}}\rvert = 24, \lvert R_{\texttt{Frescobaldi}}\rvert = 40$				Corelli	Frescobaldi	
Corelli - Frescobaldi (J48 Decision Trees)	96.9	0.05	0.97	22	2	Corelli
				0	40	Frescobaldi

Table B.3: Binary Classification - Disparate Composers. Binary classification used 10 fold cross validation, is used.

Table B.4: Binary Classification - Stylistically affiliated Composers. A decision tree ($J48$) classifier, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Mozart}} \rvert = 148, \lvert R_{\texttt{Bach}} \rvert = 126$				Mozart	Bach	
Mozart - Bach	94.2	0.06	0.94	142	6	Mozart
				10	116	Bach
$\lvert R_{\texttt{Mozart}} \rvert = 148, \lvert R_{\texttt{Haydn}} \rvert = 241$				Mozart	Haydn	
Mozart - Haydn	58.1	0.44	0.57	50	98	Mozart
				65	176	Haydn
$\lvert R_{\texttt{Joplin}} \rvert = 20, \lvert R_{\texttt{Sousa}} \rvert = 10$				Sousa	Joplin	
Sousa - Joplin	76.7	0.24	0.78	15	5	Sousa
				2	8	Joplin

Table B.5: Binary Classification - Stylistically affiliated Composers. Logistic regression, with 10 fold cross validation, is used.

	Correctly Classified (%)	Mean Absolute Error	Weighted Average F-Measure	Confusion Matrix		
$\lvert R_{\texttt{Mozart}}\rvert = 148, \lvert R_{\texttt{Bach}}\rvert = 126$				Mozart	Bach	
Mozart - Bach	91.6	0.08	0.92	135 10	13 116	Mozart Bach
$\lvert R_{\texttt{Mozart}}\rvert = 148, \lvert R_{\texttt{Haydn}}\rvert = 241$				Mozart	Haydn	
Mozart - Haydn	64.5	0.44	0.62	51 41	97 200	Mozart Haydn
$\lvert R_{\texttt{Joplin}}\rvert = 20, \lvert R_{\texttt{Sousa}}\rvert = 10$				Sousa	Joplin	
Sousa - Joplin	80	0.21	0.8	16 2	4 8	Sousa Joplin

Bibliography

[Abd11] S. Abdoli, “Iranian traditional music Dastgah classification,” *ISMIR*, pp. 275–280, 2011.

[AC11] C. Ariza and M. S. Cuthbert, *The music21 stream: A new object model for representing, filtering, and transforming symbolic musical structures*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2011.

[Akm01] A. Akmajian, *Linguistics: An introduction to language and communication*. The MIT press, 2001.

[APT+07] I. Antonopoulos, A. Pikrakis, S. Theodoridis, O. Cornelis, D. Moelants, and M. Leman, “Music retrieval by rhythmic similarity applied on Greek and African traditional music,” *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007*, pp. 297–300, 2007.

[Ass06] U. G. Assembly, “Convention on the rights of persons with disabilities,” *New York*, 2006.

[ATTB91] S. Arom, M. Thom, B. Tuckett, and R. Boyd, *African polyphony and polyrhythm: Musical structure and methodology*. Cambridge university press Cambridge, 1991.

[BAG+12] H. A. Breinbauer, J. L. Anabalón, D. Gutierrez, R. Cárcamo, C. Olivares, and J. Caro, "Output capabilities of personal music players and assessment of preferred listening levels of test subjects: Outlining recommendations for preventing music-induced hearing loss," *The Laryngoscope*, vol. 122, no. 11, pp. 2549–2556, 2012.

[Bai93] E. Bailie, *Grieg: A graded practical guide*. London: Valhalla, 1993.

[BAK+10] H. Bokil, P. Andrews, J. Kulkarni, S. Mehta, and P. Mitra, "Chronux: A platform for analyzing neural signals," *Journal of Neuroscience Methods*, July 2010.

[BB09] E. Bursztein and S. Bethard, "DeCAPTCHA: Breaking 75% of eBay audio CAPTCHAs," in *Proceedings of the 3rd USENIX conference on Offensive technologies*. USENIX Association, 2009.

[BBF+10] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, "How good are humans at solving CAPTCHAs? A large scale evaluation," in *Security and Privacy (S&P), 2010 IEEE Symposium on*. IEEE, 2010, pp. 399–413.

[BBN95] J. Blacking, R. Byron, and B. Nettl, *Music, culture, and experience: Selected papers of John Blacking*. University of Chicago Press, 1995.

[BBP$^+$11] E. Bursztein, R. Bauxis, H. Paskov, D. Perito, C. Fabry, and J. C. Mitchell, "The failure of noise-based non-continuous audio captchas," in *Security and Privacy (S&P)*, May 2011.

[BC71] A. S. Bregman and J. Campbell, "Primary auditory stream segregation and perception of order in rapid sequences of tones," *Journal of experimental psychology*, vol. 89, no. 2, p. 244, 1971.

[BC09] D. Bahanovich and D. Collopy, "Music experience and behaviour in young people," *University of Hertfordshire, UK*, 2009.

[BD73] A. S. Bregmanf and G. L. Dannenbring, "The effect of continuity on auditory stream segregation," *Perception & Psychophysics*, vol. 13, no. 2, pp. 308–312, 1973.

[BD87] G. E. P. Box and N. R. Draper, *Empirical Model Building and Response Surfaces*. John Wiley & Sons, 1987.

[Bea07] M. W. Beauvois, "Quantifying aesthetic preference and perceived complexity for fractal melodies," *Music perception*, vol. 24, no. 3, pp. 247–264, 2007.

[Ber70] L. Bernstein, *The infinite variety of music*, ser. A plume book. New American Library, 1970.

[Ber71] D. E. Berlyne, "Aesthetics and psychobiology," 1971.

[Ber76] L. Bernstein, *The unanswered question: Six talks at Harvard.* Harvard University Press, 1976.

[BGH+95] S. Buldyrev, A. Goldberger, S. Havlin, R. Mantegna, M. Matsa, C.-K. Peng, M. Simons, and H. Stanley, “Long-range correlation properties of coding and noncoding DNA sequences: Genbank analysis,” *Physical Review E*, vol. 51, no. 5, p. 5084, 1995.

[BGP06] J. P. Burkholder, D. J. Grout, and C. V. Palisca, *A history of Western music*, 7th ed. New York: W. W. Norton, 2006.

[BP78] A. S. Bregman and S. Pinker, “Auditory streaming and the building of timbre,” *Canadian Journal of PsychologyRevue canadienne de psychologie*, vol. 32, no. 1, p. 19, 1978.

[BR75] A. S. Bregman and A. I. Rudnicky, “Auditory segregation: Stream or streams?” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 1, no. 3, p. 263, 1975.

[Bre78] A. S. Bregman, “Auditory streaming is cumulative,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 4, no. 3, p. 380, 1978.

[Bre94] ——, *Auditory scene analysis: The perceptual organization of sound.* MIT press, 1994.

[Bro81] M. Brown, “Mozart and after: The revolution in musical consciousness,” *Critical Inquiry*, vol. 7, no. 4, pp. 689–706, 1981.

[Bro91] D. Brown, *Human universals*. McGraw-Hill, 1991.

[BS12] R. M. Bryce and K. B. Sprague, "Revisiting detrended fluctuation analysis," *Scientific reports*, vol. 2, 2012.

[BSEHS88] F. Benestad, D. Schjelderup-Ebbe, W. H. Halverson, and L. B. Sateren, *Edvard Grieg: The man and the artist.* University of Nebraska Press, 1988.

[BT99] L. Balkwill and W. F. Thompson, "A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues," *Music perception*, pp. 43–64, 1999.

[BvALH00] M. Blum, L. von Ahn, J. Langford, and N. Hopper, "The CAPTCHA project (completely automatic public Turing test to tell computers and humans apart)," Tech. Rep., 2000, http://www.captcha.net.

[BW] G. S. Bozarth and F. W. "Brahms, Johannes". Grove Music Online. Oxford Music Online. Oxford University Press. Accessed on 14 Jul. 2013. [Online]. Available: http://www.oxfordmusiconline.com/subscriber/article/grove/music/51879pg6

[CA10] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *Proceedings of the International Symposium on Music Information Retrieval*, vol. 11, 2010, pp. 637–642.

[CAF11] M. S. Cuthbert, C. Ariza, and L. Friedland, "Feature extraction and machine learning on symbolic music using the music21 toolkit," pp. 387–392, 2011.

[cap] Applications of CAPTCHAs. [Online]. Available: http://www.google.com/recaptcha/captcha

[Car00] N. Carnovale, *George Gershwin: A bio-bibliography*. Greenwood Publishing Group, 2000, no. 76.

[CDCDT$^+$05] O. Cornelis, R. De Caluwe, G. De Tré, A. Hallez, M. Leman, T. Matthé, D. Moelants, and J. Gansemans, “Digitisation of the ethnomusicological sound archive of the Royal Museum for Central Africa (Belgium),” *IASA journal*, no. 26, pp. 35–43, 2005.

[Cho07] P. Chordia, “A system for the analysis and representation of bandishes and gats using Humdrum syntax,” in *Proc. of the Frontiers of Research in Speech and Music Conference (FRSM 2007)*, 2007.

[Coh62] J. E. Cohen, “Information theory and music,” *Behavioral Science*, vol. 7, no. 2, pp. 137–163, 1962.

[com13] (2013, May) comScore Release April 2013 U.S.. onlinevideo rankings. Accessed July 2013. [Online]. Available: http://www.comscore.com/Insights/Press_Releases/2013/5/comScore_Releases_April_

[Cro01] I. Cross, “Music, cognition, culture, and evolution,” *Annals of the New York Academy of Sciences*, vol. 930, no. 1, pp. 28–42, 2001.

[CSC13] R. Crawford, W. J. Schneider, and N. Carnovale. (2013, July) Gershwin, george. Grove Music Online.

http://www.oxfordmusiconline.com/subscriber/article/grove/music/47026. Oxford Music Online. Oxford University Press. Accessed July 7, 2013.

[Dar97] C. J. Darwin, “Auditory grouping,” *Trends in cognitive sciences*, vol. 1, no. 9, p. 327, 1997.

[DB76] G. L. Dannenbring and A. S. Bregman, “Effect of silence between tones on auditory stream segregation,” *The Journal of the Acoustical Society of America*, vol. 59, no. 4, pp. 987–989, 1976.

[def] Movement. The Oxford Dictionary of Music, 2nd ed. rev. Ed. Michael Kennedy. Oxford Music Online. Oxford University Press. Accessed July 3, 2013. [Online]. Available: http://www.oxfordmusiconline.com/subscriber/article/opr/t237/e7024

[Den18] F. Densmore, “Teton Sioux music (Bureau of American Ethnology bulletin 61),” *Washington, DC*, 1918.

[dG] Deutsche Grammophon. Accessed May 24, 2013. [Online]. Available: http://www.deutschegrammophon.com

[DM77] M. De Montaigne. (1877) Essays. Gutenberg. Translated by Charles Cotton, and Edited by William Carew Hazlitt. Accessed on July 7, 2013. [Online]. Available: http://www.gutenberg.org/files/3600/3600.txt

[DRL+06] D. Delignieres, S. Ramdani, L. Lemoine, K. Torre, M. Fortes, and G. Ninot, "Fractal analyses for short time series: A re-assessment of classical methods," *Journal of Mathematical Psychology*, vol. 50, no. 6, pp. 525–544, 2006.

[Eld89] W. Elders, *New Josquin edition (Amsterdam, 1989-).* Vereniging voor Nederlandse Muziekgeschiedenis, 1989, vol. 30.

[Ell09] D. Ellis, "Robust landmark-based audio fingerprinting," *Online Serial*, vol. 4, May 2009. [Online]. Available: http://labrosa.ee.columbia.edu/ dpwe/resources/matlab/fingerprint

[elv] Electronic locator of vertical interval successions (ELVIS). Accessed July 13, 2013. [Online]. Available: http://elvis.music.mcgill.ca/

[Fra91] J. H. Fraser, *The American billboard: 100 years.* HN Abrams, 1991.

[Fri12] J. P. Friedlander. (2012) 2011-2012 U.S.. year-end industry shipment and revenue statistics. Online. Recording Industry Association of America. Accessed July 2013. [Online]. Available: http://76.74.24.142/4A176523-8B2C-DA09-EA23-B811189D3A21.pdf

[FS03] N. Ferguson and B. Schneier, *Practical cryptography.* Wiley New York, 2003, vol. 141.

[Fuc03] A. Fuchs, *Nonlinear dynamics in complex systems.* Springer, 2003.

[Gar78] M. Gardner, "White and brown music, fractal curves and one-over-f fluctuation," *Mathematical Games, Scientific American, New York*, 1978.

[GH08] E. Gómez and P. Herrera, “Comparative analysis of music recordings from western and non-western traditions by automatic tonal feature extraction,” 2008.

[Gib95] S. E. Gibert, *The music of Gershwin*. New Haven: Yale University Press, 1995.

[GM82] S. Goldwasser and S. Micali, “Probabilistic encryption and how to play mental poker keeping secret all partial information,” in *Proceedings of the fourteenth annual ACM symposium on Theory of computing*. ACM, 1982, pp. 365–377.

[Goo89] I. J. Good, “Surprise indexes and p-values,” *Statistical Computation and Simulation*, vol. 32, pp. 90–92, 1989.

[gov12] Accessed August 2012, We the People: U.S.. Government Petitions, https://petitions.whitehouse.gov.

[GS01] G. Grimmett and D. Stirzaker, *Probability and random processes*. Oxford University Press, 2001.

[GTM95] D. Gilden, T. Thornton, and M. Mallon, “1/f noise in human cognition,” *Science*, vol. 267, pp. 1837–1839, 1995.

[HBS65] H. E. Hurst, R. P. Black, and Y. M. Simaika, *Long-term storage: An experimental study*. Constable, 1965.

[HFH$^+$09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: An update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[HH90] K. J. Hsü and A. J. Hsü, “Fractal geometry of music,” *Proceedings of the National Academy of Sciences*, vol. 87, no. 3, pp. 938–941, 1990.

[HH91] K. J. Hsü and A. Hsü, “Self-similarity of the ‘1/f noise’ called music,” *Proceedings of the National Academy of Sciences*, vol. 88, no. 8, pp. 3507–3509, 1991.

[Hin52] P. Hindemith, *A composer’s world: Horizons and limitations*. Harvard University Press, 1952.

[HRQ05] P. Heydarian, J. D. Reiss, and M. Queen, “The Persian music and the Santur instrument,” pp. 524–527, 2005.

[Hur51] H. E. Hurst, “Long term storage capacity of reservoirs,” in *Trans. Am. Soc. Eng.*, vol. 116, 1951, pp. 770–799.

[Hur93] D. Huron, “The Humdrum toolkit: Software for music research,” *Center for Computer Assisted Research in the Humanities, Ohio State University, copyright*, vol. 1999, 1993.

[Hur94] ——, *UNIX tools for musical research: The Humdrum toolkit reference manual*, Center for Computer Assisted Research in the Humanities, Stanford University, CA, 1994.

[Ipe10a] P. Ipeirotis, “Analyzing the Amazon Mechanical Turk marketplace,” *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, pp. 16–21, 2010.

[Ipe10b] ——, “Demographics of Mechanical Turk,” 2010.

[Jeh05] T. Jehan, “Downbeat prediction by listening and learning,” in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*. IEEE, 2005, pp. 267–270.

[JGB+12] R. Joiner, J. Gavin, M. Brosnan, J. Cromby, H. Gregory, J. Guiller, P. Maras, and A. Moon, “Gender, Internet experience, Internet identification, and Internet anxiety: A ten-year followup,” *Cyberpsychology, Behavior, and Social Networking*, vol. 15, no. 7, pp. 370–372, 2012.

[Kar12] M. K. Karaosmanoğlu, “A Turkish makam music symbolic database for music information retrieval: Symbtr,” *Proc. Int. Society for Music Information Retrieval (ISMIR)*, 2012.

[KBM12] M. Kamalzadeh, D. Baur, and T. Möller, “A survey on music listening and management behaviours,” pp. 373–378, 2012.

[KCS08] A. Kittur, E. H. Chi, and B. Suh, “Crowdsourcing user studies with Mechanical Turk,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 453–456.

[Ker83] A. Kerckhoffs, “La cryptographie militaire,” *Journal des sciences militaires*, vol. IX, pp. 5–83, January 1883.

[Kin11] J. Kincaid. (2011, June) Shazam raises a huge round to the tune of 32 million dollars. Accessed June 29, 2013. [Online]. Available: http://techcrunch.com/2011/06/22/shazam-raises-a-huge-round-to-the-tune-of-32-million/

[KK08] A. Kittur and R. E. Kraut, “Harnessing the wisdom of crowds in Wikipedia: Quality through coordination,” in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 2008, pp. 37–46.

[KKBR$^+$01] J. W. Kantelhardt, E. Koscielny-Bunde, H. H. Rego, S. Havlin, and A. Bunde, “Detecting long-range correlations with detrended fluctuation analysis,” *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 3, pp. 441–454, 2001.

[KNH$^+$05] T. R. Knösche, C. Neuhaus, J. Haueisen, K. Alter, B. Maess, O. W. Witte, and A. D. Friederici, “Perception of phrase structure in music,” *Human Brain Mapping*, vol. 24, no. 4, pp. 259–273, 2005.

[Koo08] J. Koomey, *Turning numbers into knowledge: Mastering the art of problem solving*. Analytics Press, 2008.

[Koz92] A. Kozinn. (1992, August) John Cage, 79, a minimalist enchanted with sound, dies. New York Times. [Online]. Available: http://www.nytimes.com/1992/08/13/us/john-cage-79-a-minimalist-enchanted-with-sound-dies.html

[Kru91] C. L. Krumhansl, “Music psychology: Tonal structures in perception and memory,” *Annual review of psychology*, vol. 42, no. 1, pp. 277–303, 1991.

[Kru00] ——, “Rhythm and pitch in music cognition,” *Psychological bulletin*, vol. 126, no. 1, p. 159, 2000.

[LABB01] M. D. Lillibridge, M. Abadi, K. Bharat, and A. Z. Broder, “Method for selectively restricting access to computer systems,” Google Patents, February 27 2001, uS Patent 6,195,698.

[LCGM09] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, “Turkit: Tools for iterative tasks on Mechanical Turk,” pp. 29–30, 2009.

[LCM12] D. J. Levitin, P. Chordia, and V. Menon, “Musical rhythm spectra from Bach to Joplin obey a 1/f power law,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 10, pp. 3716–3720, 2012.

[Ler01] F. Lerdahl, *Tonal pitch space*. Oxford University Press, 2001, vol. 5, no. 3. [Online]. Available: http://books.google.com/books?id=6bxFrgVMDpsC&pgis=1

[Lia08] S. Lian, *Multimedia content encryption: Techniques and applications*. CRC Press, 2008.

[LM03] D. J. Levitin and V. Menon, “Musical structure is processed in “language” areas of the brain: A possible role for Brodmann Area 47 in temporal coherence,” *Neuroimage*, vol. 20, no. 4, pp. 2142–2152, 2003.

[Lon86] H. W. Longfellow, *Outre mer and driftwood.* Houghton, Mifflin and company, 1900, 1886, vol. 1.

[Mah96] G. Mahler, *Briefe*, 1896.

[Man67] B. Mandelbrot, "How long is the coast of Britain? Statistical self-similarity and fractional dimensions," *Science*, vol. 156, no. 3755, pp. 636–638, 1967.

[Man83] ——, *The fractal geometry of nature.* Times Books, 1983.

[Man89] ——, "Fractals and an art for the sake of science," pp. 14–21, 1989.

[Mat] Multitaper power spectral density estimate. Mathworks. Accessed July 2013. [Online]. Available: http://www.mathworks.com/help/signal/ref/pmtm.html

[MB79] S. McAdams and A. Bregman, "Hearing musical streams," *Computer Music Journal*, vol. 3, no. 4, pp. 26–60, 1979.

[MB93] S. McAdams and E. Bigand, "Thinking in sound: The cognitive psychology of human audition," in *Based on the fourth workshop in the Tutorial Workshop series organized by the Hearing Group of the French Acoustical Society.* Clarendon Press/Oxford University Press, 1993.

[MBR10] M. Marge, S. Banerjee, and A. I. Rudnicky, "Using the Amazon Mechanical Turk for transcription of spoken language," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.* IEEE, 2010, pp. 5270–5273.

[MCL$^+$07] D. Moelants, O. Cornelis, M. Leman, J. Gansemans, R. De Caluwe, G. De Tré, T. Matthé, and A. Hallez, "The problems and opportunities of content-based analysis and description of ethnic music," *International Journal of Intangible Heritage*, vol. 2, pp. 57–68, 2007.

[MDA$^+$03] S. J. Morrison, S. M. Demorest, E. H. Aylward, S. C. Cramer, and K. R. Maravilla, "FMRI investigation of cross-cultural music comprehension," *Neuroimage*, vol. 20, no. 1, pp. 378–384, 2003.

[MEKR11] M. Muller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 6, pp. 1088–1110, 2011.

[Mey56] L. Meyer, *Emotion and meaning in music*. Chicago: University of Chicago Press, 1956.

[Mey57] L. B. Meyer, "Meaning in music and information theory," *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, pp. 412–424, 1957.

[Mil00] G. Miller, "Evolution of human music through sexual selection," *The origins of music*, pp. 329–360, 2000.

[Mit97] T. M. Mitchell, *Machine learning*. New York: McGraw-Hill, 1997.

[MM03] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–134.

[moz] Grove Music Online: Mozart. Oxford Music Online. Accessed July 17, 2013.

[MRM$^+$05] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R. B. Davis, "Zipf's law, music classification, and aesthetics," *Computer Music Journal*, vol. 29, no. 1, pp. 55–69, 2005.

[MS12] W. Mason and S. Suri, "Conducting behavioral research on Amazon's Mechanical Turk," *Behavior research methods*, vol. 44, no. 1, pp. 1–23, 2012.

[mTu] "Amazon Mechanical Turk," accessed November 2012. [Online]. Available: https://www.mturk.com

[MVN68] B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM review*, vol. 10, no. 4, pp. 422–437, 1968.

[MVW$^+$03] B. Manaris, D. Vaughan, C. Wagner, J. Romero, and R. B. Davis, "Evolutionary music and the Zipf-Mandelbrot law: Developing fitness functions for pleasant music," in *Applications of Evolutionary Computing*. Springer, 2003, pp. 522–534.

[MW69] B. Mandelbrot and J. R. Wallis, "Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence," *Water Resources Research*, vol. 5, no. 5, pp. 967–988, 1969.

[Nao96] M. Naor, "Verification of a human in the loop or identification via the Turing test," 1996.

[Nas06] G. Nason, *Stationary and non-stationary time series*, ser. Statistics in Volcanology. Geological Society of London, 2006, ch. Chapter 11.

[NDW05] N. M. Norowi, S. Doraisamy, and R. Wirza, "Factors affecting automatic genre classification: An investigation incorporating non-western musical forms," in *Proceedings of the International Conference on Music Information Retrieval*, 2005, pp. 13–20.

[Net56] B. Nettl, *Music in primitive culture*. Harvard University Press, 1956.

[Net92] ——, "Ethnomusicology and the teaching of world music," *International Journal of Music Education*, no. 1, pp. 3–7, 1992.

[Ols67] H. F. Olson, *Music, physics and engineering*. Dover Publications New York, 1967.

[Pat03] A. D. Patel, "Language, music, syntax and the brain," *Nature Neuroscience*, vol. 6, no. 7, pp. 674–81, 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12830158

[Pat06] ——, "Musical rhythm, linguistic rhythm, and human evolution," *Music perception*, vol. 24, no. 1, pp. 99–104, 2006.

[PB00] A. D. Patel and E. Balaban, "Temporal patterns of human cortical activity reflect tone sequence structure," *Nature*, vol. 404, no. 6773, pp. 80–84, 2000.

[PBH$^+$] R. Pagano, M. Boyd, E. Hanley, E. Badura-Skoda, C. Hair, and G. Lazarevich. “Scarlatti”. Grove Music Online. Oxford Music Online. Oxford University Press. Accessed July 25, 2013. [Online]. Available: http://www.oxfordmusiconline.com/subscriber/article/grove/music/24708pg7

[PBH$^+$94] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, “Mosaic organization of DNA nucleotides,” *Phys. Rev. E*, vol. 49, pp. 1685–1689, February 1994.

[PCI10] G. Paolacci, J. Chandler, and P. Ipeirotis, “Running experiments on Amazon Mechanical Turk,” *Judgment and Decision Making*, vol. 5, no. 5, pp. 411–419, 2010.

[Per06] I. Peretz, “The nature of music from a biological perspective,” *Cognition*, vol. 100, no. 1, pp. 1–32, 2006.

[PG99] D. Perrot and R. O. Gjerdigen, “Scanning the dial: An exploration of factors in the identification of musical style,” *Proceedings of the Society for Music Perception and Cognition*, 1999.

[PG08] J. G. Palfrey and U. Gasser, *Born digital: Understanding the first generation of digital natives*. Basic Books, 2008.

[PGSB09] J. Palfrey, U. Gasser, M. Simun, and R. F. Barnes, “Youth, creativity, and copyright in the digital age,” *International Journal of Learning*, vol. 1, no. 2, pp. 79–97, 2009.

[Pha13] A. Pham. (2013, February) iTunes crosses 25 billion songs sold, now sells 21 million songs a day. http://www.billboard.com/biz/articles/news/1538108/iTunes-crosses-25-billion-songs-sold-now-sells-21-million-songs-a-day. Accessed July 22, 2013. [Online]. Available: http://www.billboard.com/biz/articles/news/1538108/iTunes-crosses-25-billion-songs-sold-now-sells-21-million-songs-a-day

[PHSG95] C.-K. Peng, S. Havlin, H. E. Stanley, and A. L. Goldberger, "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 5, no. 1, pp. 82–87, 1995.

[PKG] J. Pritchett, L. Kuhn, and C. H. Garrett. Cage, John. Grove Music Online. Oxford Music Online. Oxford University Press. Accessed July 23, 2013. [Online]. Available: http://www.oxfordmusiconline.com/subscriber/article/grove/music/A2223954

[PM12] D. Pascolini and S. P. Mariotti, "Global estimates of visual impairment: 2010," *British Journal of Ophthalmology*, vol. 96, no. 5, pp. 614–618, 2012.

[Pri] M. Priestley, *Non-linear and non-stationary time series analysis. 1988.* Academic Press, New York.

[Qui93] J. R. Quinlan, *C4.5: Programs for machine learning*. Morgan kaufmann, 1993, vol. 1.

[RC] N. C. Richard Crawford, Wayne J. Schneider. ”Gershwin, George.”. Grove Music Online. Oxford Music Online. Oxford University Press. Retreived on June 17, 2013. [Online]. Available: http://www.oxfordmusiconline.com/subscriber/article/grove/music/47026

[Rei08] D. Reisinger. (2008, December) Shazam adds 2 million tracks to music library. Accessed June 29, 2013. [Online]. Available: http://news.cnet.com/8301-17939_109-10113274-2.html

[Ren12] P. J. Rentfrow, “The role of music in everyday life: Current directions in the social psychology of music,” *Social and Personality Psychology Compass*, vol. 6, no. 5, pp. 402–416, 2012.

[Ric00] B. Richman, *How music fixed “nonsense” into significant formulas: On rhythm, repetition, and meaning*. MIT Press, 2000, ch. 17, pp. 301–314.

[RNC+95] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: A modern approach*. Prentice Hall Englewood Cliffs, 1995, vol. 74.

[Sap05] C. S. Sapp, “Online database of scores in the Humdrum file format,” in *ISMIR*, 2005, pp. 664–665.

[SC04] P. Simard and K. Chellapilla, “Using machine learning to break visual human interaction proofs (hips),” *Advances in neural information processing systems*, vol. 17, pp. 265–272, 2004.

[Sch73] C. Schwartz, *Gershwin, his life and music*. Indianapolis: Bobbs-Merrill, 1973.

[Sch75] A. Schoenberg, *Style and idea: Selected writings of Arnold Schoenberg*. University of California Press, 1975.

[Sch87] M. R. Schroeder, "Is there such a thing as fractal music?" 1987.

[Sch00] E. D. Scheirer, "Music-listening systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[Sch09] M. R. Schroeder, *Fractals, chaos, power laws: Minutes from an infinite paradise*. Courier Dover Publications, 2009.

[Şen11] S. Şentürk, "Computational modeling of improvisation in Turkish folk music using variable-length markov models," Masters Thesis, Atlanta, 2011.

[SF97] E. Selfridge-Field, *Beyond MIDI: The handbook of musical codes*. The MIT Press, 1997.

[sha] Shazam mobile implementations. http://www.shazam.com/music/web/getshazam.html Accessed June 29, 2013.

[Sha49] C. E. Shannon, "Communication theory of secrecy systems," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[Sha51] ——, "Prediction and entropy of printed English," *Bell system technical journal*, vol. 30, no. 1, pp. 50–64, 1951.

[sha13] (2013, February) Shazam press release. Accessed June 29, 2013. [Online]. Available: http://www.shazam.com/music/web/pressrelease.html?nid=NEWS20130225072239

[She87] R. N. Shepard, “Toward a universal law of generalization for psychological science,” *Science*, vol. 237, no. 4820, pp. 1317–1323, 1987.

[She00] R. Sherr, *The Josquin companion: Text*. Oxford University Press, 2000, vol. 1.

[SS86] G. Szamosi and G. Szamosi, *The twin dimensions: Inventing time and space*. McGraw-Hill, 1986.

[Ste04] M. Steen, *The lives and times of the great composers*, 2004.

[Str36] I. Stravinsky, *Chronicles of my life*. Oxford: Oxford University Press., 1936.

[Str70] ——, *Poetics of music: In the form of six lessons*. Harvard University Press, 1970, vol. 66.

[SW49] C. E. Shannon and W. Weaver, “The mathematical theory of communication,” *University of Illinois Press*, vol. 19, no. 7, p. 1, 1949.

[TED11] (2011) Luis von Ahn: Massive-scale online collaboration. TED Talks. Accessed June 2013. [Online]. Available: http://www.ted.com/talks/luis_von_ahn_massive_scale_online_collaboration.html

[Tho82] D. Thomson, “Spectrum estimation and harmonic analysis,” *Proceedings of the IEEE*, vol. 70, no. 9, pp. 1055–1096, 1982.

[Til08] B. Tillmann, “Music cognition: Learning, perception, expectations,” in *Computer Music Modeling and Retrieval. Sense of Sounds*. Springer, 2008, pp. 11–33.

[TKSW07] G. Tzanetakis, A. Kapur, W. A. Schloss, and M. Wright, “Computational ethnomusicology,” *Journal of interdisciplinary music studies*, vol. 1, no. 2, pp. 1–24, 2007.

[Tou05] G. Toussaint, “The geometry of musical rhythm,” in *Discrete and Computational Geometry*. Springer, 2005, pp. 198–212.

[TSHvA08] J. Tam, J. Simsa, S. Hyde, and L. von Ahn, “Breaking audio CAPTCHAs,” *Advances in Neural Information Processing Systems*, vol. 1, no. 4, 2008.

[Tur50] A. M. Turing, “Computing machinery and intelligence,” *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[vA05] L. von Ahn, “Human computation,” Ph.D. dissertation, Pittsburgh, PA, USA, 2005.

[VAB97] N. Vandewalle, M. Ausloos, and P. Boveroux, “Detrended fluctuation analysis of the foreign exchange market,” in *Econophysic Workshop, Budapest, Hungary*, 1997.

[vABHL03] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *Advances in Cryptology - NEUROCRYPT 2003*. Springer, 2003, pp. 294–311.

[vABL02] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically or how lazy cryptographers do AI," *Computer Science Department*, p. 149, 2002.

[vABL04] ——, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, no. 2, pp. 56–60, 2004.

[vABM08] L. von Ahn, M. Blum, and B. D. Maurer, "Controlling access to computer systems and for annotating media files," Google Patent, August 2008, wO/2008/091675.

[vAMM$^+$08] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[VC75] R. F. Voss and J. Clarke, "'1/f noise' in music and speech," *Nature*, vol. 258, no. 5533, pp. 317–318, November 1975.

[VC78] ——, "'1/f noise' in music: Music from 1/f noise," *The Journal of the Acoustical Society of America*, vol. 63, p. 258, 1978.

[Vos88] R. F. Voss, "Fractals in nature: From characterization to simulation," in *The Science of Fractal Images*, H.-O. Peitgen and D. Saupe, Eds. Springer New York, 1988, pp. 21–70.

[VSO] Vancouver Symphony Orchestra. Accessed June 23, 2013. [Online]. Available: http://www.vancouversymphony.ca

[Wan03] A. Wang, "An industrial strength audio search algorithm," in *Proc. Int. Conf. on Music Info. Retrieval ISMIR*, vol. 3, 2003.

[Wea48] W. Weaver, "Probability, rarity, interest and surprise," *Scientific Monthly. v67 i6*, vol. v67, pp. 390–392, 1948.

[Wea49a] ——, "The mathematics of communication," *Scientific American*, vol. 181, no. 1, p. 11, 1949.

[Wea49b] ——, "Recent contributions to the mathematical theory of communication," *The mathematical theory of communication*, vol. 1, 1949.

[WEG08] E. P. White, B. J. Enquist, and J. L. Green, "On estimating the exponent of power-law frequency distributions," *Ecology*, vol. 89, no. 4, pp. 905–912, 2008.

[Wika] Michael Tilson Thomas. Wikipedia. Accessed July 2013. [Online]. Available: http://en.wikipedia.org/wiki/Michael_Tilson_Thomas

[Wikb] John von Neumann. Wikiquotes. Accessed June 2013. [Online]. Available: http://en.wikiquote.org/wiki/Talk:John_von_Neumann

[WLY09] D. Wu, C.-Y. Li, and D.-Z. Yao, "Scale-free music of the brain," *PloS one*, vol. 4, no. 6, p. e5915, 2009.

[WMB01] L. Wallin, B. Merker, and S. Brown, "The origins of music," 2001.

[WS90] B. J. West and M. Shlesinger, “The noise in natural phenomena,” *American Scientist*, vol. 78, no. 1, pp. 40–45, 1990.

[WSI00] A. L.-C. Wang and J. O. Smith III, “WIPO publication wo 0211123a2, 7 february 2002,” Patent, 2000.

[You58] J. E. Youngblood, “Style as information,” *Journal of Music Theory*, vol. 2, no. 1, pp. 24–35, 1958.

[Zan06] D. H. Zanette, “Zipf’s law and the creation of musical context,” *Musicae scientiae*, vol. 10, pp. 3–18, 2006.

[Zip49] G. Zipf, “Human behavior and the principle of least effort,” *Addison-Wesley, Cambody Mus. Am. Arch. and Ethnol.(Harvard Univ.), Papers*, vol. 19, pp. 1–125, 1949.