DEVELOPING AND TESTING SUB-BAND SPECTRAL FEATURES IN MUSIC GENRE AND MUSIC MOOD MACHINE LEARNING

Fabi Prezja

Master's Thesis

Music, Mind & Technology

Department of Music, Art and Culture Studies

7 November 2018

University of Jyväskylä

JYVÄSKYLÄN YLIOPISTO

Tiedekunta – Faculty Humanities	Laitos – Department Music, Art and Culture Studies
Tekijä – Author Fabi Prezja	
Työn nimi – Title Developing and testing sub-band spectral features in music genre and music mood machine learning	
Oppiaine – Subject Music, Mind & Technology	Työn laji – Level Master's Thesis
Aika – Month and year November 2018	Sivumäärä – Number of pages 114

Tiivistelmä – Abstract

In the field of artificial intelligence, supervised machine learning enables us to try to develop automatic recognition systems. In music information retrieval, training and testing such systems is possible with a variety of music datasets. Two key prediction tasks are those of music genre recognition, and of music mood recognition. The focus of this study is to evaluate the classification of music into genres and mood categories from the audio content. To this end, we evaluate five novel spectro-temporal variants of sub-band musical features. These features are, sub-band entropy, sub-band flux, sub-band kurtosis, sub-band skewness and sub-band zero crossing rate. The choice of features is based on previous studies that highlight the potential efficacy of sub-band features. To aid our analysis we include the Mel-Frequency Cepstral Coefficients feature as our baseline approach. The classification performances are obtained with various learning algorithms, distinct datasets and multiple feature selection subsets. In order to create and evaluate models in both tasks, we use two music datasets prelabelled with regards to, music genres (GTZAN) and music mood (PandaMood) respectively. In addition, this study is the first to develop an adaptive window decomposition method for these sub-band features and one of a handful few that uses artist filtering and fault filtering for the GTZAN dataset. Our results show that the vast majority of sub-band features outperformed the MFCCs in the music genre and the music mood tasks. Between individual features, sub-band entropy outperformed and outranked every feature in both tasks and feature selection approaches. Lastly, we find lower overfitting tendencies for sub-band features in comparison to the MFCCs. In summary, this study gives support to the use of these sub-band features for music genre and music mood classification tasks and further suggests uses in other content-based predictive tasks.

Asiasanat – Keywords

Music information retrieval, music genre classification, music mood classification, sub-band features, polyphonic timbre, spectral features, adaptive spectral window decomposition

Säilytyspaikka – Depository	
Muita tietoja – Additional information	

Acknowledgements

I would like to express my gratitude to Petri Toiviainen and Pasi Saari for their outstanding supervision and support. I would like to thank, Iballa Burunat and Vinoo Alluri for helping me understand how Matlab works and answering my ‘newbie’ questions. Thank you to Valeri Tsatsishvili and Martin Hartmann, that personally and through their work helped me understand how to approach this study. Thank you to Marc Thompson and Markku Pöyhönen for their help and readiness to aid, even on a short notice! Thank you to my MMT, MT, University and ESN friends for the wonderful discussions, activities, thoughts and joys we have been sharing with one another. Thank you to my friends and family back in Greece and to the Koios family, Marousa Protopapadaki, Xristos Perdikakis, Agapi Tsatsi, Aggeliki Kouzi and Giorgos Tsaousis for their superb personal and professional help. Thank you to the staff of YTHS and KSSHP for helping me with my muscle problems when I was in trouble! Thank you to the department of music at the University of Jyväskylä for accommodating the lectures, people and activities that facilitated my academic and personal development. Lastly, words cannot express my gratitude towards my parents, Fatmira & Biku, and to Laura Immonen…

CONTENTS

Dedicated to my parents and Laura, whose support and encouragement has been incredible and unconditional…

1 INTRODUCTION

The music industry has drastically changed since the 1990s, a revolution brought upon from digital audio formats, device mobility, and computational affluence has created a need for automatic large-scale music organization and user-based predictions. Currently, music discovery and distribution is often, and at times entirely made through the world wide web, manually or automatically. Unlike earlier decades, a plethora of artists focus on distributing their music in digital formats and content provider services like Spotify, Pandora, iTunes, and even YouTube.

Music information retrieval (MIR) plays a crucial role in developing applications and tools that meet the new and developing digital music content demands. Since the 2000s MIR applications began to play an essential role in music recommendation. As a result, artists lacking the promotional benefits of record labels became more accessible and visible via automatic recommendation systems. Spotify is a prominent hub of such examples; the on-demand content service employs a plethora of MIR tasks for music big data, such as personal playlists auto-generation, music content recommendation, music meta-data association, and more. We can deduce the importance of big data for said tasks from Spotify's purchase of the 'Echo Nest[1]' database. The Echo nest data consist of over 3 million indexed artists and more than 38 million indexed songs, currently, the most extensive music and music meta-data

[1] http://the.echonest.com/ (Retrieved 15.12.2017)

database in the world. The per music track information maintained by the Echo Nest is extensive (e.g., tempo, key, time signature, timbre, similar artists).

The problem of automatic genre and mood classification focuses on the detection of music genres and music moods from the music content itself. That is without the use of expert annotators and listeners in the prediction stage. These applications have ever-increasing popularity as the need for fast and effortless digital music organization continues to grow. Categorizing music media according to emotional content and artistic style is essential to help users optimize their music exploration to other factors other than 'basic' meta-data information or manually crafted tags.

Despite the development of music genre and music mood recognition systems for more than a decade, the two applications had a ‘slow roller-coaster’ progression regarding evaluation. The focal point in understanding why development has been fundamentally slow pertains the concepts that such systems are tasked to learn. When considering music style/genre and music mood there are fundamental difficulties in consistently and reliably describing genre and mood concepts. Moreover, even if descriptions may appear consistent, the music content itself may not carry the extra-musical, contextual and cultural information that may be relevant for description. Thus, the machine learning of said descriptions becomes problematic. As a rough analogy, we can say that the closer genre and mood descriptions are to the content of music, the less ambiguous the machine learning task of such concepts may become.

One factor used in describing both music genres and music moods is timbre. The ASA defines it as "that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar". Notwithstanding the commendable attempts in defining timbre negatively (what timbre is not), we do not find an analytic proposal of what timbre is. Thus, it becomes clear that we are considering one of the most ill-defined concepts in music. Despite the ambiguity, timbre qualities have a significant role in the rapid recognition of music and sound identities, for example, avoiding the sound of a speeding car and recognizing familiar voices. Timbral features have had a considerable efficacy in music genre classification on numerous occasions. In music mood recognition, timbral features have been shown to have a supporting role amidst various feature categories, such as rhythm-based and tonality-based features. Only a handful attempts were made in modelling music mood

with timbre only features, in contrast to the general approach (a multitude of feature categories). The underlying design of many timbral features often uses the audio spectrum as the basis for feature computation. Often such timbral features are referred to as spectral features, as is also the case for this study's feature sets.

The goal of this study is to explore the performance of six timbre-based music features in music genre and music mood recognition. Five of the features belong to a family of sub-band features devised by Alluri and Toiviainen (2010). We evaluate each sub-band feature against one of the most common spectral features in MIR and speech recognition, the Mel-Frequency Cepstral Coefficients (Mermelstein, 1976). This study contains the most extensive collection of elliptical filterbank based sub-band features to be evaluated in music genre and music mood classification. In addition, the study is the first attempt in evaluating these sub-band features in music mood classification. We construct numerous classification models that we analyze and compare on par with other relevant indicators. On an exploratory basis, we also evaluate individual feature and model dimension importance for each classification task.

The next chapter focuses on the essential background, literature review and state of the art systems in music genre and music mood classification. Chapter 3 elaborates our research methodology and experimental set-up, including dataset collection, classifier set-up and data-pre-processing. Chapter 4 details our classification results with various feature selection sets in both music mood and music genre tasks. Chapter 5 focuses on the discussions of our findings along with the relevant limitations. Additionally, Appendix A enlists all classification models and classification accuracies obtained from our experiments.

2 BACKGROUND

2.1 Music Information Retrieval

Music information retrieval (MIR) is an interdisciplinary science that addresses information retrieval tasks for music and music-related content. It has a critical role in helping to develop applications and tools that meet the new and developing digital music content demands. The principal MIR applications for music content are those of recommendation, automatic classification, automatic transcription, automatic generation, and signal or instrument separation. MIR mainly engages the disciplines of computer science, electrical engineering, musicology and psychology. From within each discipline, some fields are further relevant, namely; digital signal processing, machine learning, computational intelligence, data mining, human perception, psychoacoustics and music psychology. Although MIR is relatively young, in the past decade MIR research has been rapidly expanding the outreach and performance of its applications.

2.1.1 MIREX

The MIR evaluation exchange (MIREX) is a contest that began as an initiative to standardize and systematize MIR research. MIREX serves as a platform for the incremental development of MIR tasks. The principal organizer of the contest is IMIRSEL at the University of Illinois, USA. The contest began in 2004 and had been running for 13 consecutive years, as of 2016 the total number of tasks evaluated amount to twenty-six. Evaluation tasks that pertain audio

content are numerous, for example, automatic music mood, genre and composer identification, music similarity and retrieval, melody extraction, singing voice separation, audio fingerprinting, real-time audio to score alignment and automatic drum transcription.

2.2 Feature-Based Music Concept Machine Learning

MIR has a key focus on automatic genre and mood recognition ever since the early periods of the field. The general idea behind such automatic music concept classification tasks is to attempt to model via machine learning, music concepts (genres, sub-genres, moods, etc.). The concept modeling process is often performed directly from audio examples of such concepts. Ideally, the chief expectation is that the final machine-learned model could generalize and automatically recognize the learned concepts from new music content not used during the machine learning stage. Music concept machine learning requires a collection of music examples and their related concept semantic descriptions, often referred to as 'labels.' Labels are developed and provided by human experts such that each concept (e.g., mood, genre) becomes semantically linked to each music example. Importantly, each music example is typically explained by numeric quantities referred to as 'descriptors' or 'features' (Knees & Schedl, 2013; Provost & Kohavi, 1998). Features represent shared qualities between music audio files and enable detailed representations of musical and sonic properties that are not always directly evident from the files.

2.2.1 Music Feature Abstraction Levels

Music features get extracted from raw audio files with feature extraction algorithms typically handcrafted to extract features numerically and in vector form. In MIR we find three levels of feature abstractions, low, mid and high. Each level is typically analogous to musical meaningfulness. A high-level feature stands to represent a musical concept that can be perceivable by humans. One example is that of the perceptually validated feature, Pulse Clarity (Lartillot, Eerola, Toiviainen, & Fornari, 2008). Pulse clarity numerically describes the perceived 'clarity' or 'apparentness' of the rhythmic pulse. Antithetically, a low-level feature, is lower or closer to the signal domain, such features are rarely if ever interpretable. To exemplify, consider the statistical moments of a signal (Peeters, Giordano, Susini, Misdariis, & McAdams, 2011), although statistically informative they can be perceptually

perplexing. Finally, mid-level features are often a mix of low-level features that integrate high-level concepts attempting to be perceptually relevant (Knees & Schedl, 2013).

2.2.2 Dataset Pre-Processing

The notion of data pre-processing refers to the procedures performed before feature extraction and machine learning. Data pre-processing is a crucial step for addressing dataset faults that can interfere and compromise the validity of a machine learning model.

2.2.3 Feature Pre-Processing

The idea of feature pre-processing refers to the procedures performed to extracted features before machine learning. Feature pre-processing is quite common and may include, dimensionality reduction methods, automatic redundant feature elimination and fault checking.

2.2.4 The Semantic Gap

The ‘semantic gap’ is an expression used to describe the variance in subjective interpretations for a given semantic concept or connotative meaning. In music studies and MIR (Alluri, 2012; O. Celma, 2010; Ò. Celma, Herrera, & Serra, 2006) the semantic gap regularly occurs in the process of labelling music. To exemplify, consider the semantic labels ‘Rock,’ and ‘Pop-Rock,’ numerous human listeners attributing these labels to a pool of music material might interpret the labels differently. The difference in interpretation will thus result in label to music associations that are inconsistent. This phenomenon tends to occur naturally because many concepts and connotations do not have absolute definitions and can vary culturally. In MIR, attempts to minimize the ‘gap’ often consist of majority label selection after independent annotations.

2.3 Timbre

According to the online etymology dictionary, the term 'timbre' originated from old and modern French. In modern French it is defined as 'quality of sound' [2] but in old French as the 'sound of a bell.'[2] The American National Standards Institute defined timbre as: "that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar." Alluri (2012) offers a broader definition as "the property that allows listeners to categorize and stream sound information and thereby form a mental representation of one's surroundings." In addition, timbre has been explored in terms of source identification (Handel, 1995; McAdams, 1993; Mcadams & Giordano, 2014) verbal emotion mediation (Juslin & Laukka, 2003; Laukka, Juslin, & Bresin, 2005; Scherer & Oshinsky, 1977) and non-verbal emotion mediation (Belin, Fillion-Bilodeau, & Gosselin, 2008; Bradley & Lang, 2000).

In consideration, the ANSI definition suggests that pitch is one element of timbre, it has been long-standing that this is not the case. To simplify and counter-act the definition, consider the case of a snare drum (Alluri, 2015). A snare drum may not always have a definite pitch; however, one could still differentiate one from another bearing the same loudness and pitch. Importantly, the ASA definition along with regular attempts to correct for it (Dowling & Harwood, 1986; Pratt & Doak, 1976) are negative definitions. Negative definitions maintain a degree of ambiguity since they do not detail or prescribe any specific timbral features. Between several definition attempts a single broadly accepted definition is difficult to formulate; this critical problem renders timbre one of the illest-defined concepts in music.

2.3.1 Timbre Paradigms

There are two paradigms of timbre, monophonic timbre, and polyphonic timbre, not to be confused with monophony and polyphony as in musical textures. Monophonic timbre refers to the timbre of individual instruments, voices or sound sources, (e.g., bassoon, guitar, viola). In contrast, polyphonic timbre refers to the emergent timbre of an ensemble of monophonic timbres or multiple layers of polyphonic timbres (e.g., emerging timbre of a metal concert, symphony, soundscape of a busy street). In practice, most timbre research has been focusing

[2] ("timbre | Origin and meaning of timbre by Online Etymology Dictionary," 2017) (retrieved 27.5.2017. from http://www. etymonline.com)

on monophonic timbre with considerably less attention to the equally important polyphonic timbre (Alluri, 2012).

2.3.2 MIR Features & Timbre Classification

In MIR timbre related qualities are often extracted using low-level feature extraction algorithms. Some prominent examples are; spectral centroid (Tzanetakis & Cook, 2002), zero – crossing rate (Gouyon, Pachet, & Delerue, 2000) spectral flux (Barbedo & Lopes, 2007) and spectral-roll off (E. Scheirer & Slaney, 1997), to name a few. Despite the plethora of timbre associated features, only a portion of them has been perceptually corelated and validated (Alluri & Toiviainen, 2010; Caclin, McAdams, Smith, & Winsberg, 2005; Marozeau & de Cheveigné, 2007).

Perceptual validation is essential when constructing perceptual timbre classification models, referred to as 'timbre spaces.' Timbre spaces are multidimensional models of perceptual timbre distances, often measured from human dissimilarity ratings of audio material normalized in pitch, duration, and loudness. Attack/rise time, spectral centroid and spectral flux, have been shown to be major psychoacoustic determinants of timbre (McAdams, Winsberg, Donnadieu, De Soete, & Krimphoff, 1995), this model is shown in figure 1.

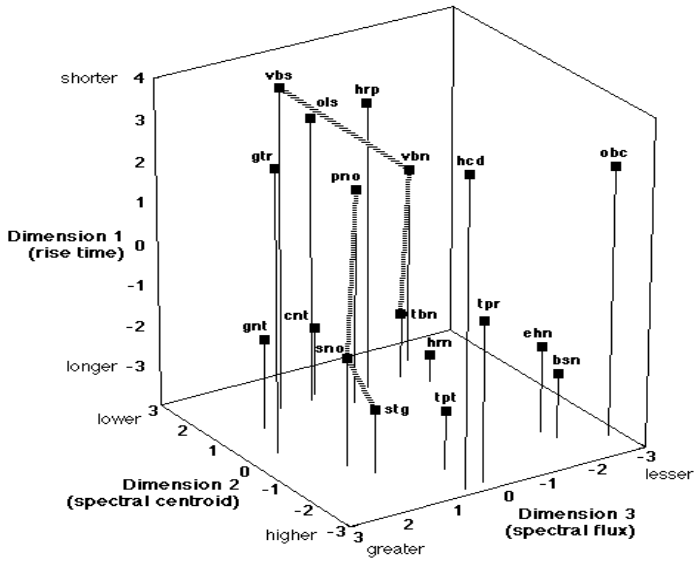

FIGURE 1. The McAdams et al., (1995) timbre space model consisted of similarity ratings between 18 synthesized instrument timbres and 'hybrid' timbres of two instruments. The dashed lines indicate hybrid timbre links to their original constituents. Some instrument code names were: french horn = hrn; trumpet = tpt; trombone = tbn; harp=hrp, Trumpar (trumpet/guitar) = tpr; oboleste(obore/celesta); vibraphone = vbs; striano (bowed string/piano) = sno; harpischord = hcd; english horn = ehn; bassoon = bsn; clarinet = cnt; vibrone

2.4 Genre and Music Genre

As found in the Online Etymology Dictionary (2017), the term ‘genre’ was first defined in 1770 as ‘a particular style of art.’ According to the Oxford English Dictionary (2016) ‘genre’ derived from French originally meaning “kind, sort, style” further stemming from the Latin term “Genus” as derived from the ancient Greek “Genos”. In the musical case, a music genre is often employed categorically and expresses a ‘style’ or ‘common group’ for a given music piece. In everyday life, music genres help to sort and refer to groups of music styles, eras, and cultural backgrounds altogether. Figure 2 highlights a portion of metal music genres, sub-genres and other genre in accordance to mutual influence.

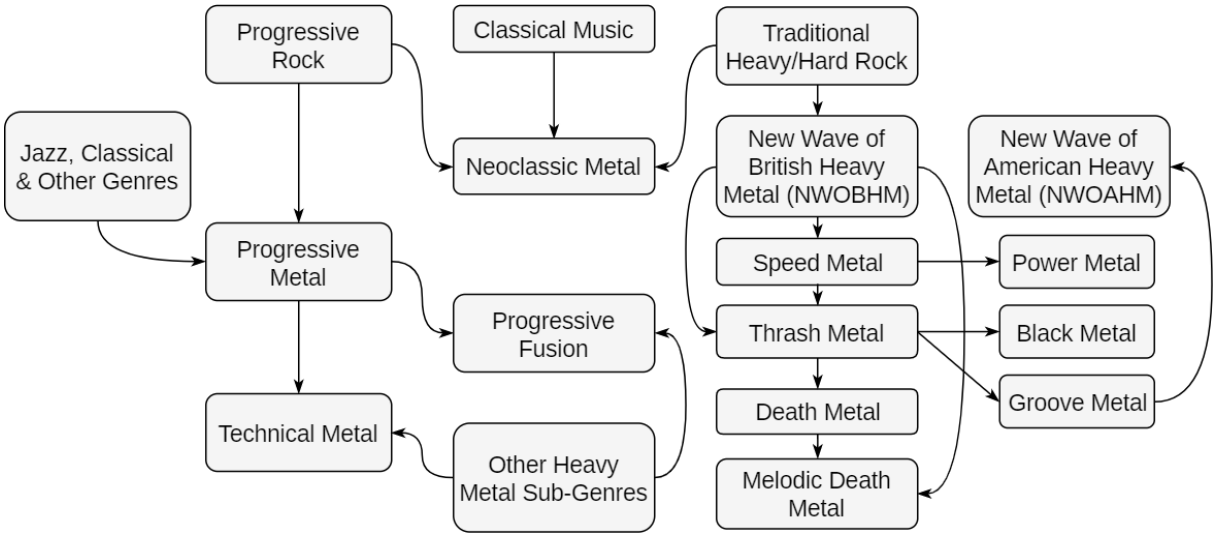

FIGURE 2. Metal genres and sub-genre influences, adopted from Tsatsishvili (2011).

Music genre is often a dynamic concept where genre membership criteria may shift in response to new cultural norms and mass re-interpretations. In general, there are no clear nor globally accepted boundaries between music genres, since the very definitions of music genres are ambiguous and subjectively inconsistent. The case is stronger for the derivative (sub-genres), or closely related genres were much cultural, structural and sonic elements may be shared. Necessarily, the semantic gap is readily pronounced in music genres, especially in new and evolving ones. Despite occasional compromises on some fundamental aesthetic 'constants' (e.g., highly distorted electric guitars in death metal), it is problematic to even consider music genres in some absolute ‘Aristotelian’ terms.

Importantly, definition inconsistencies are further manifest in novel and creative contexts. In such contexts, musicians may incorporate, modify and alternate multi-genre qualities in such a way that to describe the style a new music genre may be required altogether. Furthermore, music genres may shift within the duration of a music piece and to such an extent that one genre term is fundamentally impossible to attribute. Ultimately, precise formalization of music genres is an unattainable task, yet still, most music genres remain particularly beneficial for navigating and differentiating our music repositories.

2.5 Mood & Emotion

The terms mood and emotion often have interchangeable uses in everyday life since their differences may often seem unclear. According to the online etymology dictionary (2017), the word 'emotion' was recorded in 1650 as a "sense of strong feeling" which was generalized in 1808 to refer to any feeling. From the same dictionary, 'mood' is defined as an "emotional condition or frame of mind." Furthermore, the Oxford dictionary of psychology (2017) defines mood as "a temporary but relatively sustained and pervasive affective state." Whereas, emotion is defined as "Any short-term evaluative, affective, intentional, psychological state.'' The dictionary definitions highlight an essential contrast between the two phenomena, that of temporality. Mood was defined as a sustained affective state, but emotion was defined as a temporary affective state. To date, commonly accepted definitions of emotions and moods remains a challenging task (Frijda, 2007; Izard, 2007; Mulligan & Scherer, 2012).

Despite the definition problem, various models of emotion classification have been proposed. The dominant classification model paradigms are those of discrete and dimensional models. Discrete models are based on discrete emotion theory (P. Ekman, 1971, 1992; P. Ekman & Cordaro, 2011; P. E. Ekman & Davidson, 1994; Izard, Ackerman, Schoff, & Fine, 2000) which states that a finite set of basic emotions can be used to derive all emotions. Instead of individual emotional states, discrete models consist of various categories. Typical examples of discrete emotions are those of anger, disgust, fear, happiness, sadness, and surprise (P. Ekman, 1992).

In contrast to discrete models, dimensional models allow the mapping of emotions between dimensions in a ‘continuous-like’ space (Schlosberg, 1954; Wundt, 1907). The most popular dimensional model is Russell's (1980) circumplex model. This model maps emotions along two orthogonal dimensions, one-dimension is called ‘arousal’ and the other ‘valence.’ Each dimension has an intensity scale with a minimum and maximum value. Thayer (1990) proposed one of the most popular variants of Russell's model. Thayer’s multidimensional model maps emotions along two arousal dimensions where each dimension is also an intensity scale. Each intensity scale has one maximum value called ‘energetic arousal’ and the other called ‘tense arousal.' As described in Eerola & Vuoskoski, 2011, Thayer’s model can be superimposed to Russell’s model, figure 3 shows our adaptation of their figure.

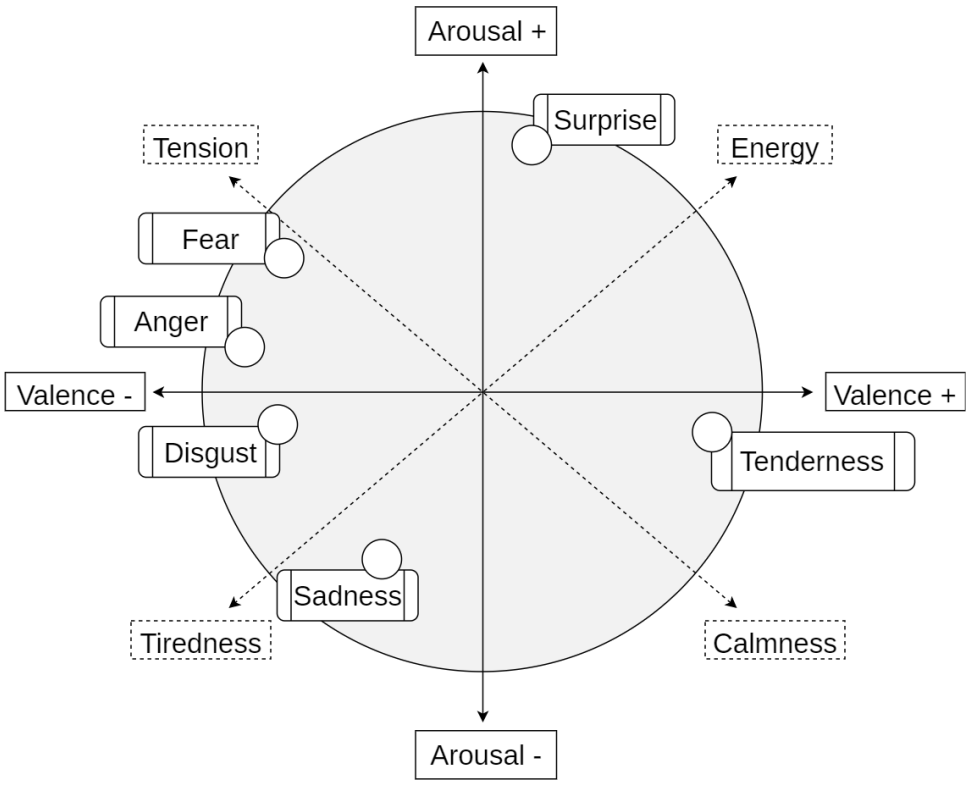

FIGURE 3. Superimposed dimensional models adopted from Eerola & Vuoskoski (2011), the dotted line stands for Thayer’s model and the straight line for Russell’s model.

2.5.1 Music & Emotion

Previous research has established that emotional reactions to music are of an uttermost importance for music-related activities (Eerola & Vuoskoski, 2013; Juslin & Laukka, 2004; Sloboda & O’Neill, 2001). The interdisciplinary field of music and emotion remains chiefly

focused on answering how and why music has such an impacting emotional effect, regardless of contextual and cultural backgrounds (Eerola & Vuoskoski, 2013). Similarly to emotion research, music and emotion research faces various criticisms and debates about the very definitions of music-induced and perceived emotions (Eerola & Vuoskoski, 2013; Juslin & Vastfjall, 2008).

According to Eerola and Vuoskoski (2013), music and emotion research utilizes four classification models (in descending order of popularity): 1) Discrete; 2) Dimensional; 3) Miscellaneous; 4) Music specific. They specify, that most discrete models employ three main categories; happiness, sadness and anger. Dimensional models on the other hand, often employ Russel's model of valence and arousal. Miscellaneous models tend to contain terms that attempt to fill the gap in categories not found in discrete and dimensional models. Finally, music-specific models share a set of common factors with dimensional models but consist of more than two dimensions.

Given the two most widespread models (discrete, dimensional), Eerola and Vuoskoski (2013) stress out two fundamental limitations. First, discrete models were mainly used with three categories that reduce and quantize the variance between emotional states. Therefore, studies that did not employ these three categories were incompatible with most of the literature that did. Second, dimensional models showcased an overreliance to the circumplex model, although studies (Bigand, Vieillard, Madurell, Marozeau, & Dacquet, 2005; Collier, 2007; Ilie & Thompson, 2006; Leman, Vermeulen, De Voogdt, Moelants, & Lesaffre, 2005) showed that valence and arousal alone are inadequate to explain the entire variance in music mediated emotions.

2.5.2 Music & Emotion in MIR.

The principal MIR music mood application is that of 'automatic mood classification' (AMC). The term 'mood' is used interchangeably to 'emotion' in MIR. In AMC, discrete emotion classification models are common because they directly satisfy the requirements of supervised machine learning. Most AMC models appear prototypically influenced by Hevner's (1936) model as adapted in figure 4. The model contained 66 adjectives arranged in 8 discrete emotion groups. Adjectives in the same group were connotatively close to each other, while geometrically opposite groups were emotionally antithetical. Further into this chapter we will

highlight all MIREX AMC datasets and their striking structural resemblance to Hevner's model.

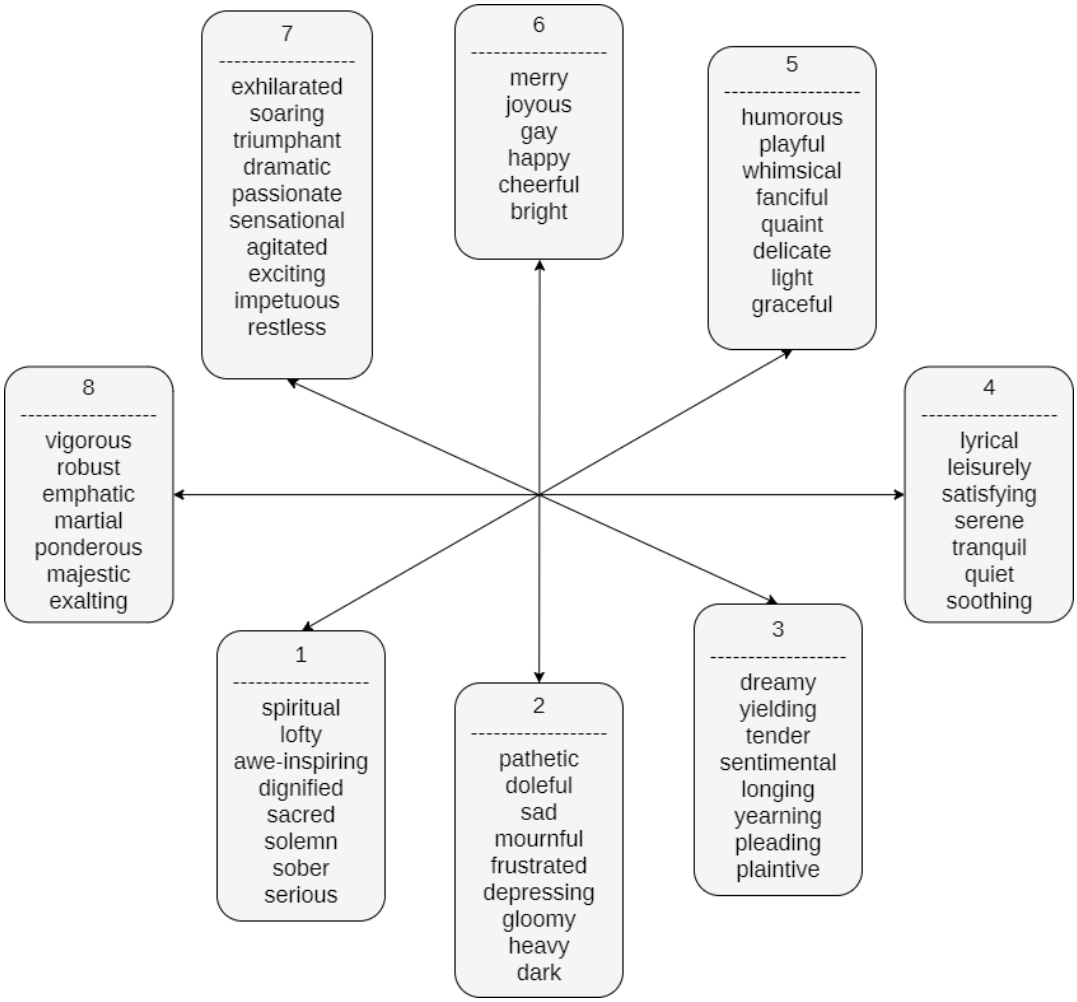

FIGURE 4. Our adaptation of Hevner's (1936) discrete emotion model, arrows connect antithetical groups of adjectives (arrows not in the original design).

2.6 Audio Signals

In this section, we will review some basic audio signal theory concepts that are relevant to our study and the literature review.

2.6.1 Periodic Signals

A periodic signal repeats itself over a given time interval; the signal is called periodic when the repetition re-occurs for equal subsequent intervals. The completion of one interval refers

to one circle and the amount of time t required to complete one circle is called a period. Let T represent a period length measured in seconds and for a continuous signal $x(t)$. Periodicity is formulated as:

$$x(t) = x(t + T)$$

We can determine the frequency f of a periodic function by keeping track of the complete circles that occur per second. We thus arrive at the following expression:

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

Where f is measured in Hertz but also expressed in radians ω as: $\omega = 2\pi f$

2.6.2 Phase

For periodic signals, the phase is measured in degrees, and as an angle, it refers to a point in the range of one complete circle. For a period $T = \frac{1}{f}$, amplitude G and phase φ of a sinusoid, the sinusoidal function $y(t)$ for the phase of any given time in $x(t)$ is:

$$y(t) = G \cdot \sin(2\pi f t + \varphi)$$

2.6.3 Amplitude

In the context of audio signals, the amplitude is a comparative measurement and refers to the strength of the atmospheric pressure with respect to mean atmospheric pressure. There are several ways to measure and represent amplitude depending on the application. Commonly, the amplitude is measured on the decibel scale (dB). The decibel is a comparative measurement of intensities, where the point of comparison of a given intensity h is the threshold of human hearing h_0 given by:

$$h_0 = \frac{10^{-12} watts}{m^2} = \frac{10^{-16} watts}{cm^2}$$

The decibel is thus defined as:

$$h(dB) = 10log_{10}\left[\frac{h}{h_0}\right]$$

Where 1 decibel (dB) is the equivalent to the 'just noticeable difference' in human auditory magnitude perception.

2.6.4 Discrete Fourier Transform

The Fourier Transform (FT) is an essential method for obtaining the frequency representation of a continuous infinite time duration signal (Bracewell & Bracewell, 1986). It is currently used for analogue system analysis and a plethora of other applications. FT has variant implementations according to signal type. For digital audio signals, the discrete Fourier Transform (DFT) implementation is often used. In MIR, the DFT is essential for understanding and generating spectral features. The DFT is typically implemented for N signal windows with the help of the Fast Fourier Transform (FFT) algorithm (Welch, 1967). The output of the DFT is a complex-valued frequency function referred to as the frequency spectrum. A conceptual spectrum analogy is that of light dispersion passing through a prism.

Formally, the DFT $X[k]$ of a signal $x[n]$ with discrete values and finite duration N, where $x[n]: n = 0, 1, \dots, N-1$. is as also of finite length such that $X[k]: k = 0, 1, \dots, N-1$. To obtain the DFT of $x[n]$ we use the following formula:

$$X[k] = \sum_{n=o}^{N-1} x[n]e^{-jk\omega_0 n} \quad k = 0,1, \dots, N-1$$

where, $j = \sqrt{-1}$, e is the natural exponent, and $\omega_0 = \frac{2\pi}{N}$. The inverse of DFT (obtaining the initial signal) from the spectra $X[k]$ is:

$$x[n] = \frac{1}{N}\sum_{n=o}^{N-1} X[k]e^{jk\omega_0 n} \quad n = 0,1, \dots, N-1$$

2.7 Machine Learning

We begin this section with an overview of the conventional machine learning methodologies and concepts. The body of work reviewed is relevant to our study which employs supervised machine learning. For this reason, we further focus on supervised learning, it's theory, and its critical considerations.

2.7.1 Background

Arthur Samuel coined the term 'machine learning' in 1959 (Samuel, 1959); it describes a vast body of knowledge within the field of Artificial Intelligence. Historically, the field originated from approaches to statistical learning and pattern recognition. Currently, the main focus is on the algorithmic learning from, and the prediction of, data. Essentially a machine learning algorithm generates a predictive model from input data. Such models can address predictive needs otherwise difficult or even impossible to achieve with conventional programming. Some popular machine learning applications are, for example, self-driving cars, automatic medical diagnosis, anomaly detection and bank loan decision support.

2.7.2 Supervised Learning

Supervised learning, or supervised classification, is a machine learning paradigm that aims at inferring a functional relationship between input and output data pairs. The input data is typically in the form of feature vectors, and the output data is a set of labels associated with the input data. For each data point, the labels are assigned by a supervising human agent or collective. A classification algorithm attempts to learn the label (output) to data (input) associations with a function. The learned function ideally would be able to generalize and predict new labels for unknown data entries. Because each new prediction relies on the learning data, each prediction is data-driven, contrary to manually developed systems where predictions may depend on programmed expert intuitions or attempts of that sort. Popular supervised learning algorithms are, for example, logistic regression, neural networks and support vector machines (Böhning, 1992; Hagan, Demuth, & Beale, 1995; Hearst, Dumais, Osuna, Platt, & Scholkopf, 1998). It is important to highlight that there is no single best learning algorithm for all problems, the 'no free-lunch' theorem provides the theoretical foundation to justify this claim (Wolpert & Macready, 1997).

2.7.3 Unsupervised Learning

Unsupervised learning focuses on discovering underlying data patterns without any supervisor labels. Unsupervised algorithms are mostly associated with the task of clustering, a process by which data structures are inferred by detecting cluster groups of potentially related data entries. A cluster typically consists of data instances that share similar feature values and thus have a relatively small distance to one another. The lack of labels in clustering methods implies that we cannot calculate an error or cost function. Popular unsupervised learning algorithms are, DBSCAN (Ester, Kriegel, Sander, & Xu, 1996), K-Means (Jain, 2010) and auto-encoders (Le, 2015), to name a few.

2.7.4 Semi-Supervised Learning

Semi-supervised learning deals with data that are partially labelled. The learning algorithm usually evaluates a significant portion of unlabeled data (analogous to unsupervised learning) along with a small portion of labelled data (analogous to supervised learning). The labelled data are critical in constructing a partial model and an essential error function. The partial model is subsequently used to assign labels to the unlabeled portion. The combination of the two is used to augment the performance of a mutual learning process. In semi-supervised learning, multiclass and one-class supervised learning algorithms are often used.

2.8 Elements of Supervised Learning

2.8.1 The Ground Truth

In supervised learning, the term ‘ground truth’ refers to all the labels devised by an expert that are true for some data. In ambiguous labelling tasks, such as music genre or music emotion labelling, objectively true labels are impossible. The difficulty lies in the inherent ambiguity and the semantic gap of the labels domain. Nevertheless, expert labelling is essential in differentiating groups of data referred to as ‘classes’. Essentially, any operation that produces a partial or complete data point (input) to label (output) association is necessary for supervised learning.

2.8.2 Training & Testing Sets

In supervised learning, it is standard practice for data to be partitioned into training and testing sets. A training set is considered as ‘known’ data, used as learning examples with which the learning algorithm builds a ‘learned’ classification model. Antithetically, a testing set consists of examples not used for training, considered ‘unknown’. The ‘unknown’ data serve to evaluate the performance of the learned classification model. The two partitions allow to determine the extent to which a final model may generalize to other data than the training data. One training and testing split is not typically enough to develop adequate confidence in a model’s generalization capacity. The limitations of one evaluation are addressed with the cross-validation partitioning method detailed later in this chapter.

2.8.3 Ground Truth Sub-Class Filtering

Ground truth sub-class filtering is a training/testing partition rule for minimizing validity issues and model sub-class overfitting. Performance inflationary effects and overfitting can occur due to the simultaneous presence of class sub-classes in the training and testing set. In music genre recognition this process is often coined ‘artist and album-filtering’ (Flexer & Schnitzer, 2009) as it targets artist and album sub-classes. To exemplify artist-filtering, let us consider we are trying to model various music genres only from audio content. If for each genre 80% of the audio examples come from one artist, we will overemphasize our learning to that artist instead of the genre they are in; consequently, the outcome will be a ‘biased’ model.

An artist filter solves this issue by restricting an artist to either the training or the testing set, in this way training and testing with the same artist is avoided. Analogously, an album filter restricts the usage of artist albums between the training and testing set, in which case the artist may be present in both partitions.

2.8.4 The Classifier Model

To elaborate on the classifier model, we will begin with an example (Luxburg & Schölkopf, 2011), let us consider a supervised learning problem with feature space $\mathcal{X}$ and label space $\mathcal{Y}$. Let us assume we are dealing with the problem of recognising a 'human' and a 'chimpanzee' based on some arbitrary genetic traits. Let $\mathcal{X}$ encapsulate the total data observations of genetic traits along multiple variables/features. Let $\mathcal{Y}$ represent which data observations in $\mathcal{X}$ belong exclusively to humans and chimpanzees. In order to learn, the algorithm will be given N such training examples, or associated data pairs $\{(X_{j,}Y_{j,})\}_{j=1}^{N}$, with $Y_j \in \{(-1,+1)\}$ where $Y_j = -1$ is the 'chimpanzee' class and $Y_j = +1$ is the 'human' class. The goal is to define a mapping $f: \mathcal{X} \rightarrow \mathcal{Y}$, that would make as few mapping mistakes of $\mathcal{X}$ to $\mathcal{Y}$ as possible. This mapping $f: \mathcal{X} \rightarrow \mathcal{Y}$ is called the classifier model and is the output of a supervised learning algorithm.

2.8.5 Model Generalization

The essential quality of a classifier model is its capacity to generalize for new unknown data, therefore model generalization is critical (Luxburg & Schölkopf, 2011). To illustrate, let us consider an arbitrary classification problem as adapted from Von Luxburg and Schölkopf (2011). The problem contains a training set of N training examples $\{(X_{j,}Y_{j,})\}_{j=1}^{N}$, by employing a learning algorithm on this data we output the classifier model called f_j. Let us now assume we have no testing set and consequently, we cannot calculate the testing error or risk of the classifier $R(f_j)$. Instead, we can only count the errors (miss-classifications) made on the training set, called the training error or empirical risk $R_{emp}(f)$. The empirical risk is thus defined as:

$$R_{emp}(f) \coloneqq \frac{1}{j}\sum_{i=1}^{j} \ell(X_i, Y_i, f(X_i))$$

Where ℓ is a 0-1 loss function defined as:

$$\ell\big(X, Y, f(X)\big) = \begin{cases} 1, & f(X) \neq Y \\ 0, & otherwise. \end{cases}$$

In the case where the empirical risk it is too large, further evaluation might not be necessary. A large empirical risk signifies that the classifier model performs unsatisfactorily with its own 'overfamiliar' training examples, which hints that it may perform even worse with 'unfamiliar' examples. In contrast, when the empirical risk is small, it is unknown how many mistakes the model would make for the rest of space $\mathcal{X}$. The rest of space $\mathcal{X}$ encapsulates unknown data that we do not possess.

In order to define a model's risk $R\big(f_j\big)$ with unknown data drawn from $\mathcal{X}$, it is common to split a dataset into training and testing sets. In this respect we obtain the two stages shown in figure 5, the training stage and the testing stage. The training stage is where the classification model is built, and the testing stage is where that model is evaluated. Ultimately, a classifier model may have the potential to generalize when the absolute divergence $\big|R\big(f_j\big) - R_{emp}(f)\big|$ is small.

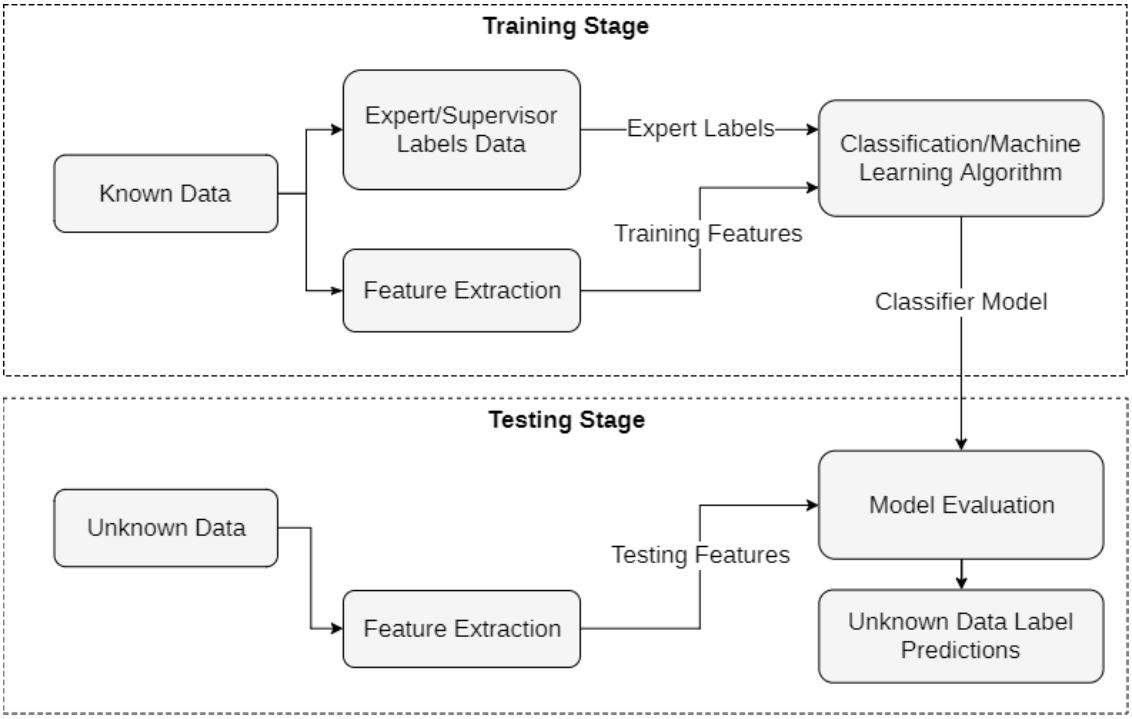

FIGURE 5. Training and testing stages in a classification pipeline.

2.8.6 Model Overfitting

Overfitting occurs when a classifier model becomes overly complex and too well fitted to its training data. Consequently, abnormalities in the learning stage (noise and random errors) are emphasized in the learned model. Ultimately, overfitting will produce an extensive number of parameters in the learning stage. This leads to a model with minimal training error $R_{emp}(f)$, but no-to insignificant generalization prospects, which means the divergence $\left|R(f_j) - R_{emp}(f)\right|$ is large.

To exemplify, let us consider the exaggerated regression case in figure 6, adapted from Von Luxburg and Schölkopf (2011). We have recorded empirical observations $n = 5$, where $(x_{1,}y_{1,}), \ldots., (x_{n,}y_{n,}) \in \mathcal{X} \times \mathcal{Y}$ and $\mathcal{X} = \mathcal{Y} = \mathbb{R}$. There are two fitted models to consider, the dashed line model f_{dashed} and the straight-line model $f_{straight}$. The f_{dashed} model is noisy and non-linear, antithetically $f_{straight}$ is linear. The f_{dashed} model has a training error = 0 while the $f_{straight}$ model has an arbitrary small training error.

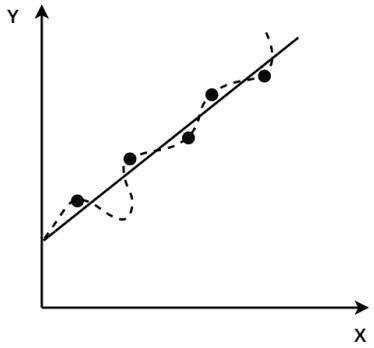

FIGURE 6. Two-model regression example.

Let us consider the true risk of both models, $R(f_{straight})$ and $R(f_{dashed})$, we know that it is not possible to access the true risk of either of them because we do not possess any testing data. In this case, which model should we prefer? Depending on the goal we must consider what constitutes as good performance, what is the state of the art for the given problem? Considering $n = 5$, more data and domain knowledge may allow us to devise a testing set to evaluate our models. Ultimately, we want to avoid both overfitting and underfitting (opposite of overfitting), since either phenomenon will undermine our models. Ideally, once we obtain

testing data, we can focus on selecting the model that manifests the lowest $\left|R(f_j) - R_{emp}(f)\right|$ (overfitting indicator) and $R(f_j)$ (testing error).

Addressing overfitting and underfitting issues can be especially challenging when the predictive task is vaguely formulated. Intuitively increasing the dimensionality of a feature space to better approximate the underlying function may be tempting, but this would mean that we also increase the chances of overfitting as more noise and random effects may be added to our model. In such cases feature selection or dimensionality reduction techniques such as Principal Component Analysis (PCA) (Wold, Esbensen, & Geladi, 1987) can help to avoid overfitting. Extending our approach to model selection would further increase our chances to select the best model. In such cases cross-validation and cross-indexing (Saari, 2009) can be effective given that we also pay close attention to the divergence value $\left|R(f_j) - R_{emp}(f)\right|$ since it is good indicator of overfitting.

2.8.7 Figures of Merit

To measure the quality of model predictions, we need to employ figures of merit (FoM). These figures are quality metrics and are key to understanding classification performance. To understand the metrics, first, we need to detail the different prediction types that can occur for any classifier model. To exemplify, let us use the previous binary class example of 'Humans' recognition against 'Chimpanzee'. We thus consider N data points $\{(x_{j,}y_{j,})\}_{j=1}^{N}$ with genetic trait observations $x_j \in \mathbb{R}^n$ and corresponding ground truth labels $y_j \in \{(-1, +1)\}$, in table 1 we show all prediction types for this binary class problem.

TABLE 1. Prediction types in classification.

Prediction Type	**Description**
Positive (P)	For x_j with class $y_j = +1$ (data of human genetic traits)
Negative (N)	For x_j with class $y_j = -1$ (data of chimpanzee genetic traits)
True Positive (TP)	Occurs when x_j with class $y_j = +1$ (human) is indeed predicted as having class $y_j = +1$. (human)
True Negative (TN)	Occurs when x_j with class $y_j = -1$ (chimpanzee) is indeed predicted as having class $y_j = -1$ (chimpanzee)
False Negative (FN)	Occurs when x_j with class $y_j = +1$(human) is predicted as having the other class $y_j = -1$ (chimpanzee)
False Positive (FP)	Occurs when x_j with class $y_j = -1$ (chimpanzee) is predicted as having the other class $y_j = +1$ (human)

2.8.8 Classification Accuracy

Classification Accuracy (CA) is the most common figure of merit for classification tasks; it is the proportion of successful predictions against all predictions:

$$CA = \frac{TP + TN}{TP + FN + FP + TN} = \frac{Correctly\ Predicted}{All\ Predictions}$$

This metric helps us to access the goodness of a model with respect to its predictive power. Importantly, a single accuracy score by itself does not inform us on model overfitting potential. All MIR classification tasks feature CA as their primary figure of merit.

2.8.9 K-Fold Cross-Validation

K-fold cross-validation (KFCV) is a data partition method used for validating models and reducing overfitting (Kohavi, 1995). In practice, the entire dataset is partitioned into K folds and iterated K times, for each iteration, one fold is used as a testing set while the remaining folds are combined into one training set. To exemplify, consider the case shown in figure 7 where $K = 4$, for that example during the first iteration the first fold is used as a testing set and folds; 2,3,4 are aggregated as a training set. Any evaluation metric can be used to score each iteration if the metric is classification accuracy (CA), then the final KFCV score is the average CA across iterations. A key benefit of KFCV is that, when all iterations have been evaluated, the entire dataset has been used both for training and testing.

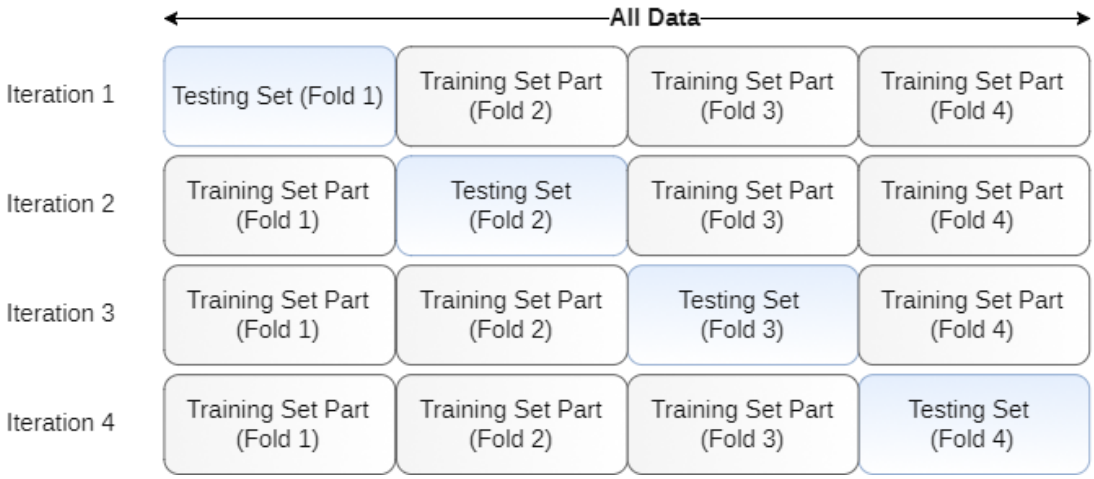

FIGURE 7. Four-fold cross-validation scheme.

2.9 MIREX

For more than a decade, music genre and music mood classification systems have been competing in the MIREX contest. MIREX classification tasks are evaluated in supervised learning with various music datasets/sub-tasks. Participants submit their classification systems which are evaluated with task-specific guidelines. After evaluation, the competing systems get ranked according to a specified figure of merit.

Structure

We begin our review with the evaluation guidelines of both 'Automatic Genre Classification' (AGC) and 'Automatic Mood Classification' (AMC), excluding that of 2006 since no such tasks where evaluated. After the guideline reviews, we continue by individually analyzing each music genre and music mood sub-task. Each sub-task review consists of three parts; 1) Dataset analysis 2) Top system evolution (on a yearly basis) 3) State of the art (SoA) analysis 4) General trends found in top performing systems.

2.9.1 MIREX Evaluation Guidelines (2005 – 2017)

Every year the MIREX contest publishes task evaluation guidelines with specifications for cross-validation, significance testing, performance metrics and tasks specific requirements (artist filter, hierarchical ground truths, etc.). All participating systems are evaluated with these guidelines and are ranked based on a performance metric, typically classification accuracy. Feature extraction, training and classification times are also measured for an independent run-time ranking. Overall, the evaluation guidelines in AGC and AMC have been varying throughout the runtime of the tasks but in 2009 both task guidelines became identical (excluding task-specific requirements). Table 2 maps the aforementioned changes for AGC and AMC on a yearly basis; Currently, both AGC and AMC guidelines specify 3-fold cross-validation, classification accuracy (CA), and significance testing with Friedman's ANOVA and the Tukey-Kramer HSD (honestly significant difference).

Guidelines:	**Task Specific**	**Cross-Validation**		**Significance Testing**			**Figure of Merit**
AGC Year	Artist Filter	3 – Fold Cross Validation	5 – Fold Cross Validation	Friedman's ANOVA	McNemar's Test	Tukey-Kramer HSD	Classification Accuracy
2005	No	Yes	Yes	No	Yes	No	Yes
2007	Yes	Yes	No	No	Yes	No	Yes
2008	Yes	Yes	No	Yes	Yes	No	Yes
2009	Yes	Yes	No	Yes	No	Yes	Yes
2010 - 2017	Yes	Yes	No	Yes	No	Yes	Yes

TABLE 2. Evaluation guidelines for AGC.

Guidelines:	**Task Specific**	**Cross-Validation**		**Significance Testing**			**Figure of Merit**
AMC Year	Artist Filter	3 – Fold Cross Validation	5 – Fold Cross Validation	Friedman's ANOVA	McNemar's Test	Tukey-Kramer HSD	Classification Accuracy
2007	-	-	-	Yes	No	Yes	Yes
2008 - 2017	-	Yes	No	Yes	No	Yes	Yes

TABLE 3. Evaluation guidelines for AMC.

2.9.2 MIREX AGC Review (2005 – 2017)

Historically, automatic genre classification systems have been developed from MIDI since 1997 by Dannenberg, from audio by Matityaho and Furst (1995) and later popularised by Tzanetakis and Cook (2002). For more than a decade, music genre classifications systems have been developing and improving the automatic recognition of music content into music genres. The MIREX community introduced the genre classification task in 2005 which is currently running for 11 years.

AGC Sub-Tasks & Datasets

In 2005 the first AGC sub-task had two music datasets, 'Magnatune' and 'USPOP'. Magnatune contained 1515 whole length audio files from 9 genres with a hierarchical g**r**ound truth. The USPOP dataset included 1414 whole length audio files from 6 genres. USPOP was used for single level classification and Magnatune for hierarchical classification (dropped after 2005). In 2007 MIREX introduced a new sub-task/dataset called 'mixed genre', it contained 7000, 30-second excerpts equally drawn from 10 genres. In 2008 the 'Latin genre' sub-task and dataset (Silla Jr, Koerich, & Kaestner, 2008) were introduced. The new dataset

contained 3160 songs distributed in 9 genres and aimed at facilitating the recognition of popular Latin and dance Latin songs. 'K-POP Genre' was the latest sub-task/dataset introduced in 2014 by IMIRSEL and KETI. The dataset contained 1894, 30-second excerpts of Korean popular music unevenly allocated in 7 genres (J. H. Lee, Choi, Hu, & Downie, 2013; Lie, 2012). Table 4 shows each sub-task and evaluation period along with relevant dataset properties (genre labels, audio format, etc.).

TABLE 4. AGC sub-task and dataset properties.

Evaluation Year:	**2005**		**2007 - 2017**	**2008 - 2017**	**2014 - 2017**
Sub-Task:	Audio genre classification		Mixed popular genre classification	Latin genre classification	K-Pop genre classification
Sub-Task:	Magnatune	USpop	Mixed Genre	Latin Genre	K-Pop Genre
Genre Labels:	Blues Classical Electronic Ethnic Folk Jazz Newage Punk Rock	Electronica/Dance Newage Rap/Hip-Hop Reggae Rock	Blues Classical Country Dance Jazz Metal Rap Hip Hop Rock and Roll Romantic	Bachata Bolero Forro Gaucha Merengu E Pagode Salsa Sertaneja Tango	Ballad Dance Folk Hip-Hop R&B Rock Trot
Total Classes:	9	5	10	9	7
Audio Files	1515	1414	7000	3227	1894
Length:	Unedited	Unedited	30 Seconds	Unknown	30 Seconds
Format:	. Mp3	. Mp3	. Wav	. Mp3	. Wav

MIREX AGC Sub-Task Review

In the following section, we review all MIREX AGC sub-tasks in chronological order. The general outline of the review begins by aggregating (on a yearly basis) all top performing submissions, their learning algorithms and feature specifications. Each sub-task review concludes with an analysis of each sub-task's state of the art along with the overall trends found amongst all top submissions. In 2005 we encounter a separate dataset and hierarchical taxonomies, we review it separately and with a slightly alternate format. The format differs in that we review the top three performing systems (instead of one) and treat the year in

isolation. In addition, the 2007 top system was unavailable and is thus excluded from our review.

Audio Genre Classification Task (2005)

In 2005 the first AGC sub-task had 15 participants, 13 of which completed the task within the given 24-hour run time. The top three classification accuracies for both the USPOP and the Magnatune datasets, where as follows: 1) 82.34% 2) 81.77% (Bergstra, Casagrande, & Eck, 2005); 3) 78.81% (Mandel & Ellis, 2006). The first two systems are identical, and their learning algorithm was AdaBoost (Freund & Schapire, 1997). The third system employed DAG-SVM which is a special multiclass case of SVM (Platt, Cristianini, & Shawe-Taylor, 2000).

The first and second system used the following features; 256 real cepstral coefficients (RCEPS), 64 Mel-frequency cepstral coefficients (MFCCs), 32 linear prediction coefficients, 32 low-frequency magnitudes, 16 spectral roll-offs, one linear prediction error, and one zero crossing rate. Each feature was extracted with a 47-millisecond window and further partitioned into 13.9-second segments of which the mean and variance were taken, resulting in 804 feature dimensions. In contrast, the third system used 20 MFCC coefficients and the maximum likelihood of fitting a Gaussian distribution to those MFCCs. We see that both systems consisted of spectral features and that they had the MFCCs in common.

Mixed Genre Sub-Task (2008 – 2017)

From 2008 until 2017 the mixed genre sub-task had 148 entries. Each year, the highest classification accuracies were as follows (chronological order): 66.41% (Tzanetakis, 2007) in 2008; 73.33% (Cao & Li, 2009); 73.64% (Seyerlehner, Schedl, Pohle, & Knees, 2010); 80.07% (Hamel, 2011); 76.13% (Wu & Jang, 2012); 76.23% (Wu & Jang, 2013); 83.55% (Wu & Jang, 2014); 76.27% (Wu & Jang, 2015); 76.84% (J. Lee & Nam, 2017a, 2017b; J. Lee, Park, Kim, & Nam, 2017; J. Lee, Park, Nam, et al., 2017). Table 5 enlists the learning features, classification algorithms and classification accuracies of these systems.

TABLE 5. Yearly top systems in the mixed genre sub-task, the highest performance is highlighted in bold.

Year	Accuracy	Learning Algorithm	Learning Features
2008	66.41%	Support Vector Machines	MFCCs, spectral centroid, roll-off, flux
2009	73.33%	Support Vector Machines	Gaussian super vector (GSV) of: MFCCs, rhythm pattern (RP).
2010	73.64%	Support Vector Machines	Spectral pattern (SP), variance delta spectral pattern (VDSP), logarithmic fluctuation pattern (LFP), correlation pattern (CP), spectral contrast pattern (SCP);
2011	80.07%	Pooled Features Classifier	Principal Mel – spectrum components (PMSC)
2012	76.13%	Support Vector Machines	2009 Features + multi-level visual features (MLVFs), beat tracking local texture representations
2013	76.23%	Support Vector Machines	2012 Features + beat-level based heterogeneity features
2014	**83.55%**	2 × Support Vector Machines	2012 Features
2015	76.27%	Support Vector Machines	2012 Features
2017	76.84%	Support Vector Machines	Deep convolutional neural network generated low level features

The state of the art was set in 2014; the top system had a dual SVM classifier set-up with GSV features, MLVFs, MFCCs, rhythm patterns and beat tracking local textures. The features were adapted to a Gaussian mixture model (GMM) derived from another GMM pre-trained from an external music dataset. The external dataset coined 'Universe Background Model' (UBM) contained 2000 music tracks randomly selected from the 7digital[3] music database. The author's GSV framework has been competing since 2009, scoring the highest in 2009, 2012 and 2013. Although subsequent feature additions facilitated performance improvement, the state of the art was identical to their 2012 system, except that it had a dual SVM classifier. Thus, it follows that the considerable rise in performance came from implementing two SVMs conjoined with confidence based late fusion. Conclusively, it seems that after the addition of GSV features the authors turned their attention to develop their classifier set-up that ultimately allowed them to achieve the highest accuracy.

[3] https://github.com/7digital/python-7digital-api (Retrieved 12.10.2018)

Considering all entries in table 5, we see that most systems employed at least one SVM classifier and either partially or solely included spectral features. For most entries, the STFT was the primary spectrum source which highlights a common approach to spectrum acquisition. In addition, eight out of ten entries contained MFCCs which shows another regularity between the systems. Thus, we can conclude that SVM, STFT, spectral features and MFCCs have had the highest consistency throughout the runtime of the sub-task.

Latin Genre Sub-Task (2008 – 2017)

Before we enlist the top systems in the Latin genre sub-task, we note that the top systems reported in 2009, 2010, 2011, 2014, and 2017 were also top for the concurrent mixed genre sub-task. In addition, the relevant information for the 2012 and 2015 top entries was not available and will be excluded from this review.

The Latin Genre Recognition task collectively had 123 entries, each year the highest classification accuracy was as follows (chronological order): 65.17% (Cao & Li, 2008); 74.66% with their mixed genre system (Cao & Li, 2009); 79.86% with their mixed genre system (Seyerlehner et al., 2010); 82.31% with their mixed genre system (Hamel, 2011); 77.60% (Pikrakis, 2013); 78.64% with their mixed genre system (Wu & Jang, 2014); 69.88% (Lidy & Schindler, 2016); 75.86% with their mixed genre system (J. Lee, Park, Nam, et al., 2017); Table 6 enlists the learning features, classification algorithms, concurrent sub-tasks and performances of these systems.

The state of the art was set in 2011 (Hamel, 2011); the system consisted of a pooled features classifier and Principal Mel – Spectrum Components (PMSC) features. PMSCs were made in three consecutive steps; 1) DFT acquisition 2) Mel scale compression 3) PCA whitening. Analytically, the DFT spectrum was obtained with a window of 1024 samples and a frame step of 512 samples. Next, the spectral energy bands were obtained from filtering the DFT with 256 Mel-spaced triangular filters. Ultimately, PMSC unitary variance features were obtained by employing PCA whitening. Once the feature set was exported it continued to the pooling stage. In the pooling stage, the authors applied a set of pooling functions to the PMSCs. The resulting pooled features were used with a multi-layer perceptron that consisted of one, 2000-unit layer with sigmoid activations. The combination of the pooling stage and the classifier was referred to by the authors as the Pooled Features Classifier (PFC).

TABLE 6. Yearly top systems in the Latin genre sub-task, the highest performance is highlighted in bold.

Year	Accuracy	Concurrent Top	Learning Algorithm	Learning Features
2008	65.17%	-	Support Vector Machines	Gaussian super vector (GSV) of: MFCCs
2009	74.66%	Mixed Genre	Support Vector Machines	Gaussian super vector (GSV) of: MFCCs, rhythm pattern (RP) features.
2010	79.86%	Mixed Genre	Support Vector Machines	Spectral pattern (SP), variance delta spectral pattern (VDSP), logarithmic fluctuation pattern (LFP), correlation pattern (CP), spectral contrast pattern (SCP);
2011	**82.31%**	Mixed Genre	Pooled Features Classifier (Multi-Layer Perceptron Based)	Principal Mel – spectrum components (PMSC)
2013	77.60%	-	Deep Neural Network	Self-similarity based rhythmic signatures
2014	78.64%	Mixed Genre	2 × Support Vector Machines	2009 Features + multi-level visual features (MLVFs), beat tracking local texture representations
2016	69.88%	-	2 × Convolutional Neural Networks	Mel-spaced spectrograms
2017	75.86%	Mixed Genre	Support Vector Machines	Convolutional neural network generated low level features

Amongst the systems in table 6, five out of eight contained at least one SVM classifier where the reminder entailed various types of neural networks. Six out of eight systems employed or incorporated spectral features of which three featured MFFCs. The majority specification mirrored those in the mixed genre sub-task since five out of eight systems were identical in both tasks. The state of the art system generally drifted away from the common approaches except in that it employed spectral features.

AGC K-POP Genre Sub-Tasks (2014 – 2017)

The MIREX contest founded two K-POP genre sub-tasks in 2014, the main goal was to explore the extent to which western AGC systems can classify non-western material. A dual ground truth format was devised to explore any cross-cultural effects in AGC. One ground

truth was devised by Korean annotators and the other by American annotators. We have found that in each task all top yearly systems were identical, except for 2014. For this reason, we combine both tasks (American and Korean) into one collective task review. Thus, in the following review, we refer to the Korean ground-truth systems as 'KGT' and the American ground-truth systems as 'AGT'. In addition, the 2015 top system information was not available and will be excluded from this review.

Since 2014 both K-POP genre sub-tasks collectively had 38 entries. On a yearly basis, the highest classification accuracy was as follows; (KGT) 65.58% (Seyerlehner & Schedl, 2014); (AGT) 63.25% with their mixed and Latin genre system (Wu & Jang, 2014); (KGT) 64.36%, (AGT) 62.35% with their Latin genre system (Lidy & Schindler, 2016); 67.90% (KGT), 67.74% (AGT) with their mixed and Latin genre system (J. Lee, Park, Nam, et al., 2017). Table 7 shows the ground truth version, learning features, learning algorithms and performance of these systems.

TABLE 7. Yearly top systems in the K-POP genre sub-task, the highest performance is highlighted in bold.

Year	Accuracy	Ground Truth	Concurrent Top	Learning Algorithm	Learning Features
2014	65.58%	KGT	-	Support Vector Machines	Spectral Pattern, delta spectral pattern, variance delta spectral pattern, logarithmic fluctuation pattern, correlation pattern, spectral contrast pattern, local single gaussian model, George Tzanetakis model
2014	63.25%	AGT	Mixed Genre, Latin Genre	2 × Support Vector Machines	Gaussian super vector (GSV) of: MFCCs, rhythm pattern (RP), multi-level visual features (MLVFs), beat tracking local texture representations
2016	64.36% (KGT) 62.35% (AGT)	Both	Latin Genre	2 × Convolutional Neural Networks	Mel-spaced spectrograms
2017	**67.90%** (KGT) **67.74%** (AGT)	Both	Mixed Genre, Latin Genre	Support Vector Machines	Deep convolutional neural network generated low level features

For both tasks, the state of the art was set in 2017 (Lee, et al., 2017); the system comprised of an SVM classifier and two deep convolutional networks for feature generation. The neural networks were pre-trained separately (J. Lee & Nam, 2017b; J. Lee, Park, Kim, et al., 2017). One network was trained with the Million-Song-Dataset (MSD) (McFee, Bertin-Mahieux, Ellis, & Lanckriet, 2012) and the other with the NAVER[4] dataset. The MSD had a ground truth devised from LastFM tags while NAVER had 107 genre classes after extensive data filtering. The DCNNs targeted short sample level characteristics from sample level filtered audio signals. Their system is a typical example of 'feature transfer learning' (Choi, Fazekas, Sandler, & Cho, 2017; Oquab, Bottou, Laptev, & Sivic, 2014; Yosinski, Clune, Bengio, & Lipson, 2014) where features and feature extractors are learned from one or more tasks and are used in other tasks.

In line with other AGC sub-tasks, the majority of systems employed SVM classifiers with most feature sets either including or solely comprising of spectral features. The STFT continued to be the common spectrum acquisition approach. All entries, but one, were top performing and identical in both ground truths along with other concurrent sub-tasks. The consistent performance between sub-tasks and annotation ground truths (Korean, American) may suggest that there is less 'sensitivity' to ground truth cultural effects and varying music material.

2.9.3 Audio Mood Classification (AMC)

Studies in music psychology show that music emotion plays a critical role in mediating, expressivity and artistic intent (Juslin, Karlsson, Lindström, Friberg, & Schoonderwaldt, 2006; Juslin & Laukka, 2004). Various studies (Cunningham, Bainbridge, & Falconer, 2006; Cunningham, Jones, & Jones, 2004; Vignoli, 2004) highlighting needs in seeking to organize and retrieve music based on its emotional content. The main aim of automatic mood classification systems is to facilitate and automatize this process.

[4] http://naver.com/ (Retrieved 12.10.2018)

AMC Sub-Tasks & Datasets (2007 – 2017)

The first MIREX AMC sub-task and dataset were introduced in 2007, coined 'Audio Music Mood' (AMM) the dataset contained 600, 30-second audio excerpts equally divided into five classes. The ground truth consisted of mood clusters, each employed a set of interrelated adjectives from the AMG mood repository (Hu & Downie, 2007). The audio to cluster mapping was performed by human judges from a draft audio pool. Audio that had a majority agreement between subjective classifications was maintained. The goal of mood clusters and human judges was to shrink the semantic and sociocultural mood space. Noteworthy is that the cluster architecture highly resembled Hevner (1936) discrete music emotion model.

The latest AMC sub-task/dataset coined 'K-POP Music Mood' (K-POP MM) was developed under the K-POP genre paradigm. Analogously to the K-POP genre task, the task includes two dataset instances, each with separate ground truths. The common dataset holds 1438, 30-second audio excerpts unevenly distributed into 5 clusters. The cluster adjectives were identical to the AMM dataset, but the music to cluster allocation was performed by majority agreement separately in each annotator group (American, Korean). The goal for K-POP Music Mood was to evaluate cultural effects concerning music mood classification. Table 8 aggregates the AMM and K-POP MM sub-tasks and their dataset properties.

TABLE 8. AMC sub-task and dataset properties.

Evaluation Period:	**2007 - 2017**	**2014-2017**
Sub-Task:	Audio Music Mood	K-POP Music Mood
Classes (Adjectives):	▪ **Cluster 1** (passionate, rousing, confident, boisterous, rowdy) ▪ **Cluster 2** (rollicking, cheerful, fun, sweet, amiable/good natured) ▪ **Cluster 3** (literate, poignant, wistful, bittersweet, autumnal, brooding) ▪ **Cluster 4** (humorous, silly, campy, quirky, whimsical, witty, wry) ▪ **Cluster 5** (aggressive, fiery, tense/anxious, intense, volatile, visceral)	
Total Classes:	5	
Audio Files:	600	1438
Length:	30 Seconds	30 Seconds

2.9.4 MIREX AMM Review (2007 – 2017)

Before we enlist the top AMM systems, we exclude the 2011 and 2015 entries since their specifications were not available. Sub-task concurrent top systems were found in 2009, 2013 and 2016.

Since 2007 the AMM sub-task had a total of 178 entries, yearly, the highest classification accuracies were; 61.50% (Tzanetakis, 2007); 63.67% (Peeters, 2008); 65.67% (Cao & Li, 2009); 64.17% (Wang, Lo, Jeng, & Wang, 2010); 67.83% (Paiva, 2012); 68.33% with their 2013 mixed genre system (Wu & Jang, 2013); 66.33% (Panda, Rui, & Paiva, 2014); 63.33% with their 2016 Latin and K-POP genre system (Lidy & Schindler, 2016); 69.83% (Park, Lee, Nam, Park, & Ha, 2017). Table 9 enlists the learning features, learning algorithms and performance of these systems.

TABLE 9. Yearly top systems in the AMM mood sub-task, the highest performance is highlighted in bold.

Year	Accuracy	Concurrent top	Learning Algorithm	Learning Features
2007	61.50%	-	Support Vector Machines	Spectral centroid, roll-off, flux, MFCCs
2008	63.67%	-	Gaussian Mixture Model	MFCCs, spectral flatness measure (SFM), spectral crest measure (SCM)
2009	65.67%	Mixed Genre, Latin Genre	Support Vector Machines	Gaussian super vector (GSV) of: MFCCs, rhythm pattern (RP) features
2010	64.17%	-	Ensemble Classifier (Support Vector Machines/AdaBoost)	23 Features in four groups; 1 dynamic (root mean square loudness), 11 spectral (e.g. roughness, entropy, brightness), 5 timbre (e.g. spectral flux, MFCCS), 6 tonal (e.g. key clarity, chroma peak, chroma centroid)
2012	67.83%	-	Support Vector Machines	312 Features (e.g. MFCCs, zero crossing rates, inharmonicity, loudness, timbral width, roll-off, centroid)
2013	68.33%	Mixed Genre	Support Vector Machines	2009 Features + multi-level visual features (MLVFs), beat tracking local texture representations, beat-level based heterogeneity features
2014	66.33%	-	Support Vector Machines	410 Features (expanded 2012 feature set)
2016	63.33%	Latin Genre, K-POP Genre	2 × Convolutional Neural Networks	Mel-spaced spectrograms
2017	**69.83%**	-	Support Vector Machines	Deep convolutional neural network generated low level features (taught with million song database)

The state of the art was established in 2017 (Park, Lee, Nam, et al., 2017; Park, Lee, Park, Ha, & Nam, 2017); the system consisted of one SVM classifier and deep convolutional neural network generated features. To extract all features, a DCNN was pre-trained with 100,000 audio excerpts from the 'million song dataset' (MSD). The DCNN contained five convolutional layers leading to an output layer of 5000 ‘artists’ nodes. For each artist, fifteen songs made up a learning set, three a validation set and two a testing set. Raw audio excerpts were not involved; instead 128 band Mel-spectrograms were used. The Mel-spectrograms were computed from a 1024 sample dynamically compressed STFT. The DCNN architecture consisted of ‘ReLu’ activations followed by batch normalization. Dropout was used after the final convolutional layer and before the prediction layer. The Network optimization was performed with stochastic gradient descent and Nesterov momentum.

Analogously to AGC tasks, seven out of nine systems consist of at least one SVM classifier. Eight out of nine systems include or solely consist of spectral features, seven of which further include MFCCs. All systems employ the STFT as their spectrum acquisition technique and we find ensemble and GMM classifiers in the top systems. Three systems (2009, 2013, 2016) have concurrent top accuracies in genre sub-tasks (Mixed, Latin, K-Pop).

K-POP Mood Review (2014 – 2017)

We note that all K-POP mood systems, except in 2014, were identical in both ground truths and with their concurrent top AGC systems. For this reason, we merge our review for both ground truths except for 2014. Also, the 2015 system description was unavailable and will be excluded.

Since 2014 both K-POP mood tasks had a total of 44 entries, each year’s highest classification accuracy was as follows: 62.35% with their K-POP (AGT), mixed and Latin genre system (Wu & Jang, 2014); 64.23% (Xu & Gu, 2014); 60.75% (KGT), 62.98% (AGT) with their AMM, K-POP(AGT, KGT) and Latin genre systems (Lidy & Schindler, 2016); 65.34% (KGT), 65.34% (AGT) with their K-POP (AGT, KGT), mixed and Latin genre systems (Park, Lee, Nam, et al., 2017; Park, Lee, Park, et al., 2017). Table 10 shows the ground truth version, learning features, learning algorithms and performance of these systems.

TABLE 10. Yearly top systems in the K-POP Mood sub-task, the highest performance is highlighted in bold.

Year	Accuracy	Ground Truth	Concurrent Top	Learning Algorithm	Learning Features
2014	62.35%	KGT	K-POP Genre (AGT), Mixed Genre Latin Genre	2 × Support Vector Machines	Gaussian super vector (GSV) of MFCCs rhythm pattern (RP), multi-level visual features (MLVFs) and beat tracking local texture representations
2014	64.23%	AGT	-	Support Vector Machines	Spectral pattern (SP), delta spectral pattern (DSP), spectral contrast pattern (SCP), logarithmic fluctuation pattern (LFP), correlation pattern (CP), beat-level texture (BLT), beat spectrogram (BS)
2016	60.75% (KGT) 62.98% (AGT)	Both	AMM K-POP Genre (AGT & KGT) Latin Genre	2 × Convolutional Neural Networks	Mel-spaced spectrograms
2017	**65.34%** (KGT) **65.34%** (AGT)	Both	K-POP Genre (AGT & KGT) Mixed Genre Latin Genre	Support Vector Machines	Deep convolutional network generated low level features

The state of the art for both ground truths was found in 2017 (Park, Lee, Nam, et al., 2017; Park, Lee, Park, et al., 2017); the system used a pre-trained DCNN as a feature generator and an SVM classifier. The system set up was identical to the authors' entries in K-POP, ‘Mixed’ and ‘Latin’ genre sub-tasks. The only configuration change across ground truths was the DCNN learning material. The DCNN for AGT was taught with the MSD and NAVER datasets while the KGT system was taught with the MSD dataset.

Although in this task a small number of systems was under review, we can observe that the majority followed suit to AGC and AMM. Most systems consisted of at least one SVM and either included or comprised of spectral features with the STFT as the common spectrum acquisition method. Importantly, all systems but one (2014 AGT) were top performing in other sub-tasks, which again suggests a minimal cross-cultural and classification domain restriction for these systems.

2.9.5 MIREX Limitations

Reflecting on the state of MIREX evaluation, we find five significant limitations: 1) Both AGC and AMC do not focus on reporting any training set accuracies. The training to testing accuracy divergence is essential for accessing overfitting, in its absence, it is difficult to evaluate the participating systems critically; 2) The standard deviation cross-validation accuracy is absent which makes it even more challenging to access models adequately; 3) No model robustness testing was used, by not using any such tests (e.g., noise generation, irrelevant data transformation, data augmentation, etc.) it becomes considerably challenging to reflect on model performance in other than 'ideal' conditions; 4) Classification accuracy does not control for the quality of errors made between systems. Although systems might be performing well in most cases, a commercial application may lead to an end user losing confidence when the model makes mistakes of bad quality ('irrational' confusions). This approach could also prove useful in ranking when top performance and generalization is equal between several systems; 5) Unavailability of sub-task data and several top entry specifications limits task development prospects.

2.9.6 Concurrent Top Systems

So far, we focused on individual system performance in AGC and AMC, throughout our analysis it became clear that four systems extended beyond top individual sub-task performance. These systems were: 1) Cao & Li (2009) top performing in AMM, Mixed and Latin Genre 2) Wu & Jang (2014) top performing in K-POP Mood (AGT), Mixed and Latin Genre. 3) Lidy & Schindler (2016) top performing in K-POP Genre (AGT & KGT), K-POP Mood (AGT & KGT), AMM and Latin Genre. 4) Park, Lee, Nam, et al., (2017) and Park, Lee, Park, et al., (2017) top performing in all subtasks, except for AMM. The consistencies found in the first three (1-3) were the use of spectral features and SVM classifiers. Given these observations, we can consider that certain system designs can indeed perform best between multiple sub-tasks in both AGC and AMC. It is difficult to interpret the exact reasons why these systems perform well between tasks since they follow the general trends found in each respective task. Nevertheless, it is plausible to consider that the systems mentioned above are the closest to a single system approach to music concept classification, given that they are not overfitting (that we cannot access).

2.9.7 AGC Remarks

In review of all AGC sub-tasks, we observe six principal points of interest: 1) High performances were achieved despite the presence of an artist filter; 2) SVMs and spectral features remained the ‘go to’ options for the majority of systems; 3) The majority of spectral features contained the MFCCs; 4) One system (J. Lee, Park, Nam, et al., 2017) performed the highest in every AGC sub-task; 5) Outside the majority of classifier and feature choices, the reminder of top systems employed neural networks either as a classifier or as a feature extraction method.

2.9.8 AMC Remarks

Between all AMC sub-tasks, we find that the AGC remarks 2, 3, 5 also apply to AMC. Considering this substantial similarity, we only amend the fourth point with Lidy's and Schindler's (2016) system that performed the best across all AMC sub-tasks. In line with the AGC remarks, Lidy’s and Schindler’s system demonstrated a diminished sensitivity to cultural and data specific effects.

2.9.9 Closing Remarks

Despite all the promising results, we find a relatively slow development in all sub-tasks and against a ‘glass ceiling’ (Pachet & Aucouturier, 2004) of classification accuracy. This is not surprising given the ambiguous nature of the semantic labels associated with each task. Another reason may also lie in the unmusicality and perceptual irrelevance of many popular low-level features. It is plausible to consider that transcending past the ‘glass ceiling’ might require a new approach in both feature design and systems evaluation.

3 METHODOLOGY

The outline of our methodological approach consisted of two stages, data pre-processing and data classification. In the pre-processing stage, we extracted and processed six Spectro–temporal features for two music datasets. After feature extraction, we grouped the features into combinatorial and feature selection sub-sets, predominately manually and once with a feature selection algorithm. In the data classification step, each feature group was used as an input to three learning algorithms. Both methodological stages are summarized in figure 8; this scheme is also the general outline of supervised machine learning as found in the MIREX contest. In this chapter, we elaborate on each stage shown in figure 8.

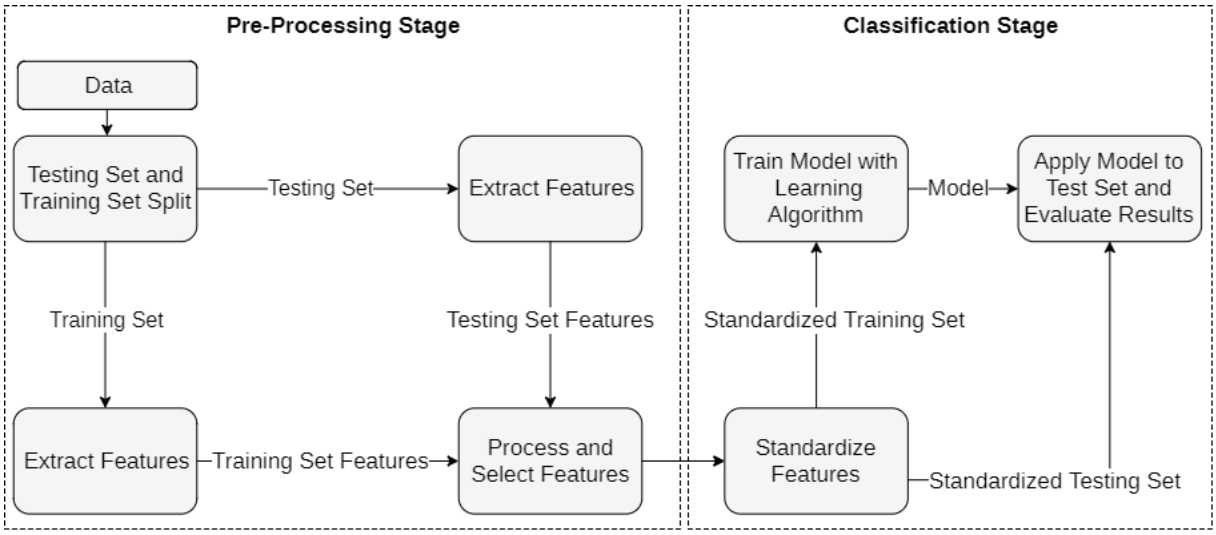

FIGURE 8. Both stages of our experimental design, each step represents one procedure with each line being the outcome of each procedure.

In our pre-processing stage, five out of six features were sub-band features (Alluri & Toiviainen, 2010; M. A. Hartmann, 2011) and were the focus of this study; the remainder was sub-band based (MFCCs) which served as a comparative baseline. All sub-band features were extracted with an independent temporal window size for each of their sub-bands. The window size is computed by an operation we call 'Filter Dependent Windowing'. This operation, which we detail later in this chapter, calculated a unique window size suited to each sub-band frequency range. Thus, the sub-band features may be rephrased as FDW based sub-band features. In the last stage of feature extraction, we statistically summarized all extracted feature vectors with their mean and standard deviation. Once all the features were extracted and summarized, we proceed to the data classification stage. In the classification stage, we performed supervised machine learning with three learning algorithms; Support Vector Machines (SVM); Multinomial Logistic Regression (MLR); K – Nearest Neighbors (K-NN).

In total, our experimental design consisted of six features, two datasets and three learning algorithms. We constructed a factorial design shown in table 11 to map the key experimental factor combinations. Furthermore, each feature was summarized with two statistics, effectively resulting in more combinations than the factorial design suggests. For this reason, further into this chapter we also map the additional feature subsets and feature selection sub-sets. Our primary focus in this study, was to rank, explore and contrast the classification accuracy of our features set when selected individually, algorithmically and all together. It is important to point out that our primary aim was not to explore all possible feature combinations; instead, we only wanted to focus on essential feature subsets for the classification tasks.

TABLE 11. The factorial design of this study, each cell is an experimental factor. Each column item is the 'input' of each column item on the right.

Factors→ Levels ↓	**Music Dataset**	**Learning Features**	**Machine Learning Algorithm**
1	GTZAN (Fault Filtered)	Sub-Band Entropy	Support Vector Machines (SVM)
2	GTZAN (Fault Filtered + Artist Filtered)	Sub-Band Skewness	Multinomial Logistic Regression (MLR)
3	PandaMood	Sub-Band Kurtosis	K – Nearest Neighbours (K-NN)
4	-	Sub-Band ZCR	-
5	-	Sub-Band Flux	-
6	-	MFCC Coefficients	-

3.1 Music Databases

We evaluated multiple feature sets between two mutually independent music datasets; these datasets served as the audio material which allowed us to extract learning features and perform machine learning. We chose two classification tasks for our evaluations, automatic music genre classification (AGC) and automatic music mood classification (AMC). Each task was associated with one dataset, AGC was evaluated with the 'GTZAN' (Tzanetakis & Cook, 2002) dataset while AMC was evaluated with the 'PandaMood' (Panda, Malheiro, Rocha, Oliveira, & Paiva, 2013) dataset. The choice of these datasets was due to the corresponding MIREX datasets not being available. For this reason, we considered to obtain datasets that satisfied four factors: 1) Similarity to the MIREX dataset; 2) Public availability; 3) Supporting background literature; 4) Adequate dataset size.

GTZAN Dataset – Automatic Genre Classification

The GTZAN dataset was first introduced by Tzanetakis and Cook (2002), and it was mainly used for music genre classification. The dataset consists of 1000 audio files in .au format; the files are grouped into ten classes (blues, classical, country, disco, hip-hop, jazz, metal, reggae, rock). Each of the classes contains 100 music excerpts with a duration of 30 seconds each. All audio files are unnamed and do not carry any relevant metadata.

The GTZAN dataset has been shown to contain audio replicas, distortions and mis-labelings (Sturm, 2012, 2013a, 2014b). These issues along with the lack of an artist filter can influence the classification procedure and effect classification performance. For this reason, we constructed a fault filtered version of GTZAN along and an artist filter specification. We specifically used the faults and artists listed by Sturm (2013b). Also, we expanded our fault filter in the case of multiple versions of a music piece; we kept only the first version of an excerpt as it appeared in each genre. Thus, the resulting fault filtered dataset contained 903 audio files, more information is shown in table 12. From here on, we will be referring to our fault filtered GTZAN simply as GTZAN and use the 'AF' term to indicate artist filtering of the fault filtered data.

PandaMood Dataset – Automatic Mood Classification

The 'PandaMood' dataset is the audio-only part of a larger multimodal dataset (Panda et al., 2013), 'PandaMood' is the code name we borrow from Sturm (2014a). The dataset is devised similarly to the MIREX 2007 mood classification dataset. It employs 903, 30-second-long mp3 audio files categorized roughly alike into five classes also called 'mood clusters'. Each cluster is numbered from 1 to 5 and contains the same tags as the MIREX 2007 mood dataset.

The corresponding tags are as follows: Cluster 1) passionate, rousing, confident, boisterous, rowdy; Cluster 2) rollicking, cheerful, fun, sweet, amiable/good-natured; Cluster 3) literate, poignant, wistful, bittersweet, autumnal, brooding; Cluster 4) humorous, silly, campy, quirky, whimsical, witty, wry; Cluster 5) aggressive, fiery, tense/anxious, intense, volatile, visceral. The source of the music was the AllMusic[5]database where the music material was fetched by selecting songs that corresponded to the MIREX 2007 mood cluster tags. Professionals tagged the fetched music material, but access to their evaluation criteria and procedural details was not made public. Further dataset information is appended to table 12 below.

TABLE 12. GTZAN & PandaMood dataset properties.

Music Dataset→ Properties↓	**GTZAN**	**PandaMood**
Classification Task:	Music Genres	Music Moods
Classes (Percentage of total files):	• Blues (11.1%) • Classical (10.6%) • Country (10.6%) • Disco (10.1%) • Hip-Hop (9.7%) • Jazz (9.3%) • Metal (9.6%) • Pop (8.7%) • Reggae (9.3%) • Rock (10.9%)	• Cluster 1: passionate, rousing, confident, boisterous, rowdy; (18.8%) • Cluster 2: rollicking, cheerful, fun, sweet, amiable/good natured; (18.2%) • Cluster 3: literate, poignant, wistful, bittersweet, autumnal, brooding; (23.8%) • Cluster 4: humorous, silly, campy, quirky, whimsical, witty, wry; (21.2%) • Cluster 5: aggressive, fiery, intense, tense/anxious, volatile, visceral (18.1%)
Number of Classes:	10	5
Length of excepts:	30 seconds	30 seconds
Audio files count:	903	903

[5] https://www.allmusic.com/ (Retrieved 12.10.2018)

3.2 Feature extraction (Pre-processing Stage)

In this section, we elaborate on our feature extraction strategies for each feature used in our study. The feature extraction step was responsible for generating our learning features from the audio content. Each feature represented a quantitative measure of a particular spectral property over time. In this study, we extracted six spectro-temporal features: 1) Sub-Band Entropy (SB-Entropy); 2) Sub-Band Flux (SB-Flux); 3) Sub-Band Kurtosis (SB-Kurtosis); 4) Sub-Band Skewness (SB-Skewness); 5) Sub-Band Zero Crossing Rates (SB-ZCR); 6) Mel-frequency Cepstral Coefficients (MFCCs). We first elaborate on the sub-band features and conclude by elaborating on the baseline MFCC features. We implemented the entire feature extraction process with the MIRtoolbox 1.6.1 (Lartillot, Toiviainen, & Eerola, 2008) in the MATLAB environment. Figure 9 shows the general overview of our feature extraction set up, including statistical summarization.

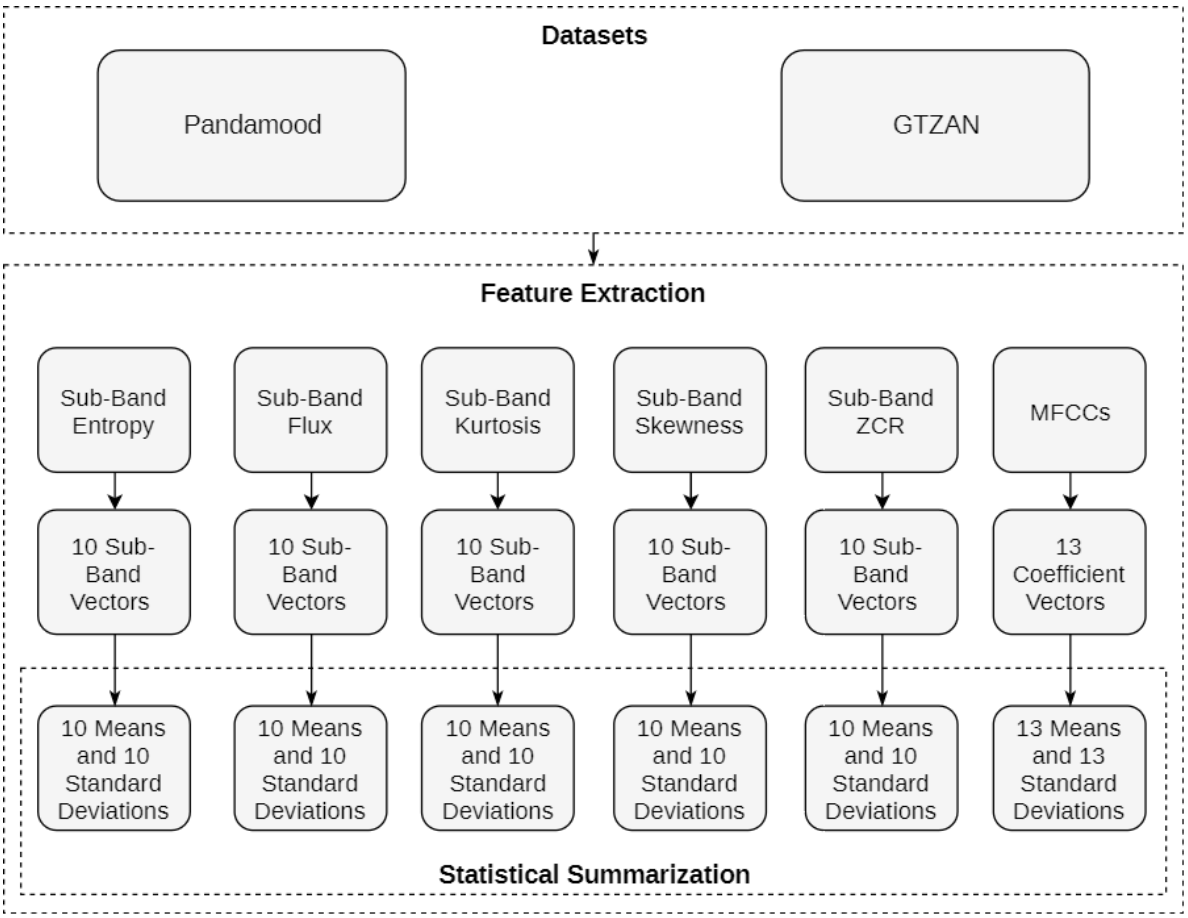

FIGURE 9. The general overview of our feature extraction step.

3.2.1 Sub-Band Feature Generation

In our study, five out of six features were from the family of sub-band/multiresolution features. The broad criteria for qualifying any feature as a sub-band feature is that feature computations are applied to a filter-bank decomposed signal. The main difference between a sub-band feature and a 'broadband' feature, is that the latter typically consists of only one feature vector computed for the entirety of the input frequency range. In contrast, a sub-band feature is a collection of multiple vectors computed from different segments of the input frequency range. The term sub-band or multiresolution stems from the operation of filter-bank decomposition as the frequency range is 'partitioned' into smaller frequency bands.

Our procedural flow to generate the filter dependent windowing (FDW) sub-band features consisted of four steps:1) Filter bank-decomposition; 2) Signal window decomposition with FDW; 3) Spectrum extraction; 4) Feature computation. In more detail: 1) A signal enters filter-bank decomposition which results in a set of ten sub-band signals; 2) The new set of signals enters the filter dependent windowing procedure (FDW), this results in each sub-band signal to be windowed with a unique sub-band based window size; 3)A short-time Fourier transform (STFT) spectrum is computed for each window and every sub-band signal. 4)A feature is computed for each spectrum window of every sub-band; In our case, this procedure generated a spectral sub-band feature consisting of 10 sub-band feature component vectors. Figure 10 details the extraction pipeline specific to our study, we elaborate each operation in the proceeding sections.

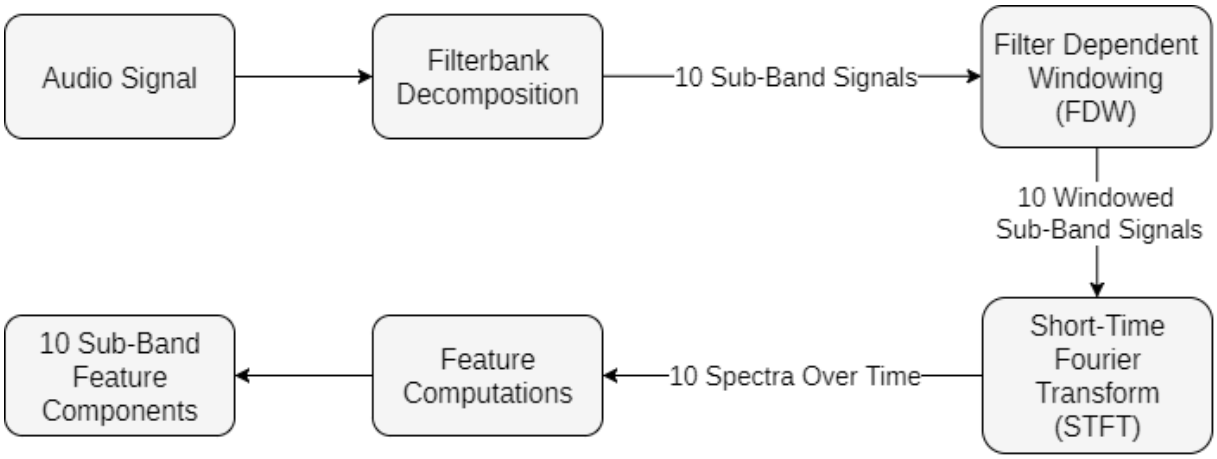

FIGURE 10. The FDW sub-band spectral feature generation pipeline for one audio file, each arrow indicates the outcome of each operation.

Filterbank Decomposition

The sub-band feature extraction procedure begins by implementing filter-bank decomposition. This process is the fundamental building block for our feature set; it generates the necessary sub-bands for which we compute our spectral features. In general, filter-bank decomposition tries to imitate the procedure by which the ears cochlea analyses incoming vibrations in the frequency domain. The filter-bank design we implement is based on previous work that introduced and evaluated the sub-band flux feature (Alluri & Toiviainen, 2010; M. A. Hartmann, 2011). The two studies share a filter-bank design based on Scheirer's (1998) design that was used for beat extraction and tempo analysis. The main difference between the designs is the filter order; Scheirer's (1998) design used an order of six while proceeding designs used an order of two.

In our design, we use ten non-overlapping, octave range, fourth order elliptical filters as our filter-bank. The design comprises of 10 filters of three types: one low pass filter, eight bandpass filters and one high pass filter. Each filter/sub-band covers a unique octave range of frequencies, the number of filters is dependent on the sampling rate, we employ a sampling rate of 44.1 Khrz which thus requires 10-octave size sub-bands to cover the full frequency range. Figure 11 presents Alluri's and Toiviainen's (2010) filter-bank frequency response with a filter order of two. Table 13 highlights each sub-band frequency range and corresponding octave range.

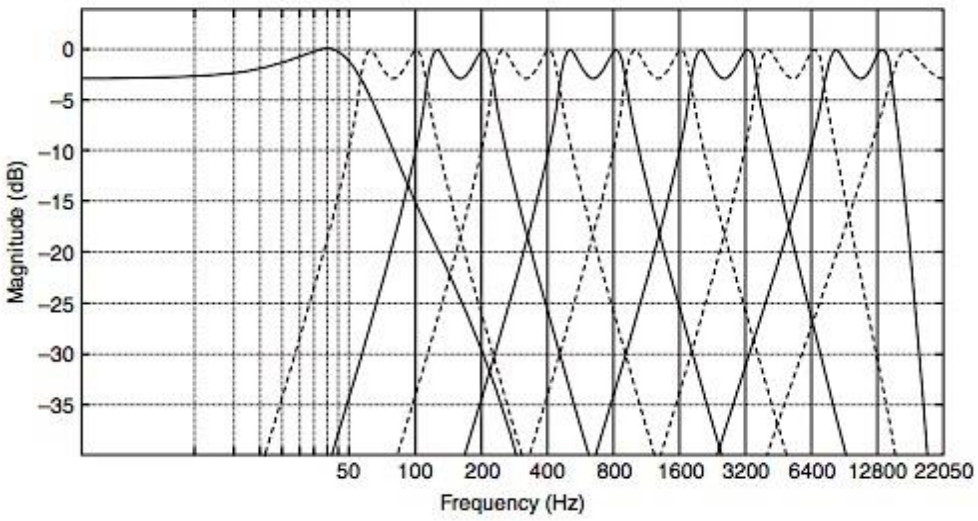

FIGURE 11. The aggregate and each sub-band filter frequency response (Alluri & Toiviainen, 2010).

TABLE 13. Frequency range and octave range for each sub-band filter.

Sub-band Filter No	Frequency Range (Hertz)	Octave Range (Each note + 35 cents)
1	0 – 50	G1
2	50 - 100	G1 – G2
3	100 - 200	G2 – G3
4	200 - 400	G3 – G4
5	400 - 800	G4 – G5
6	800 - 1600	G5 – G6
7	1600 - 3200	G6 – G7
8	3200 - 6400	G7 – G8
9	6400 - 12800	G8 – G9
10	12800 - 22050	G9

3.2.2 Filter Dependent Windowing

To window our sub-band features, we developed a windowing method we refer to as 'Filter Dependent Windowing' (FDW). With this method, we pre-computed and applied unique window sizes for each filter in our filter-bank. The central concept is to adapt the window amount and size to the frequency range of each filter/sub-band. The method aims in enriching the statistical summaries (mean, standard deviation) by extracting adaptive sizes and thus, varying amounts of windows for each sub-band frequency band. Consequently, each extracted sub-band signal vector will be of a different length. With FDW we obtain an dynamic analogy between frequency range and window size (figure 13), windows are larger for the lower sub-bands (frequency length is larger) and much smaller for the highest ones (frequency length is smaller). Instead of a one size fits all window size, FDW was developed to adopt the window size to best suit the frequency content of each filter. This process was fundamentally inspired by the wavelet transform (Daubechies, 1990) method. Figure 12 highlights the procedural pipeline of FDW.

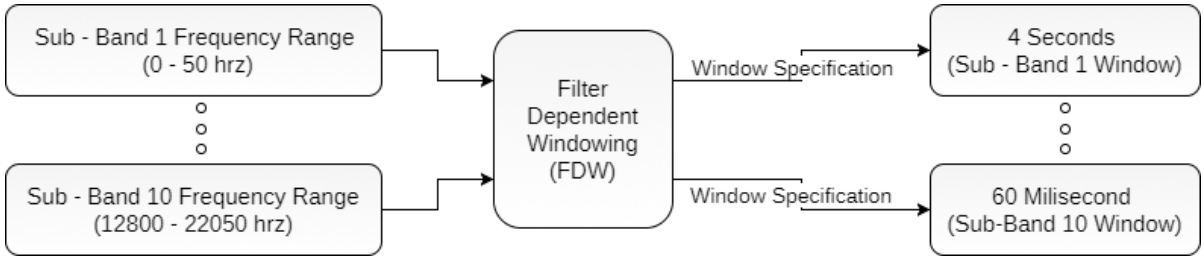

FIGURE 12. The FDW procedural pipeline, the inputs are sub-band specifications and the output are window size specifications.

To obtain the window length size of each sub-band in our filter-bank w_j with $j = 1 \ldots 10$ we used the following function:

$$w(j) = \frac{100}{f(c_j)}$$

Where $w(j)$ results in the window length specification measured in seconds and $f(c_j)$ is the central frequency of the j th sub-band. One generalized form of the central frequency calculation we used, is the following:

$$f(c_j) = \begin{cases} \sqrt{(f_{h,j}\, f_{l,j})}\ , & if\ \ \frac{f_{h,j}}{f_{l,j}} \geq 1.1 \\ \frac{1}{2}(f_{h,j+}\, f_{l,j}), & if\ \ \frac{f_{h,j}}{f_{l,j}} < 1.1 \end{cases}$$

Where $f_{h,j}$ is the high-pass cut off frequency and $f_{l,j}$ is the low-pass cut off frequency of the jth sub-band. This form can be used with varying filterbank specifications, including ours. By implementing the FDW with our filter-bank specification, we obtained the window sizes show in table 14, the hop/overlap size for each window was 50%, meaning that each new window began from the central temporal location of the previous window.

TABLE 14. Sub-band filter-bank components, frequency ranges, central frequencies and the corresponding FDW window size.

Sub-band Filter (Index)	Frequency Range (Hertz)	Central Frequency (Hertz)	FDW -Window Size (Seconds)
1	0 - 50	25	4
2	50 - 100	71	2.92
3	100 - 200	141	4.91
4	200 - 400	283	2.46
5	400 - 800	566	1.23
6	800 - 1600	1131	0.61
7	1600 - 3200	2263	0.31
8	3200 - 6400	4525	0.15
9	6400 - 12800	9051	0.08
10	12800 - 22050	16800	0.06

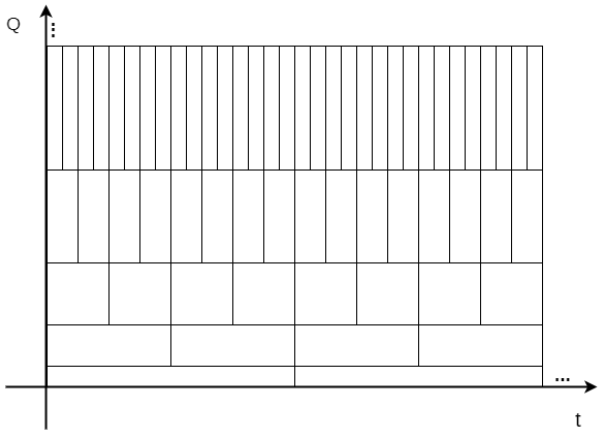

FIGURE 13. A visual analogy of the resulting FDW windows over time t for every sub-band index Q. We can discern that as the sub-band index increases the window length decreases.

3.2.3 Spectrum Computation

In our study, we compute the spectrum over N signal windows with the help of the short-time Fourier transform (STFT). The resulting STFT windows become the basis for which feature computation takes place. The STFT is one of the fundamental components of most timbre features because it is the input of feature computation.

3.2.4 DFT Window Function

In the case of finite duration signals such as ours, it is standard practice to apply a window function for each window before the DFT. The reason is to avoid spectral leakage, which introduces unwanted frequency components that did not exist in the DFT input. Spectral leakage occurs when the sampling of an infinite periodic signal with period N is not an integer multiple of the period of that signal. The DFT causes the frequency components of such a signal to swift which result to either discontinuities or overlaps when the signal repeats. The DFT window function is a function that has a non-zero value only for some interval. In our implementation we select the Hann discrete window, defined as:

$$W[n] = sin^2\left(\frac{\pi n}{N-1}\right)$$

Where for some timeseries $X[i]: i = 0,1,\dots,N-1$, we obtain $G[k] = W[n] \cdot X[i]$ in the time domain.

3.3 Sub-Band Spectral Features

For every window and every sub-band, the STFT of each FDW window was obtained before each feature set was computed. This cascade of operations yields our final sub-band features. Sub-band flux is adopted from (Alluri & Toiviainen, 2010) which served as the foundation for developing the rest of the sub-band features. To the best of our knowledge SB-Entropy, SB-ZCR, SB-Kurtosis and SB-Skewness based on Alluri's & Toiviainen's (2010) specification have not been introduced before. In the following section, we elaborate on the details of each sub-band feature computation. In the final part of this section, we also describe the baseline MFCC feature extraction process.

3.3.1 Sub-Band Entropy

The idea of entropy, and particularly information entropy has its roots in information theory (Shannon, 2001). It was introduced as a metric of uncertainty, information and choice that allows the estimation of the average minimum bits of information in a message. In physics and mainly statistical mechanics, entropy corresponds to the amount of 'disorder' in a system. Sub-band entropy has had varying uses mainly in automatic speech recognition and analysis (Egenhofer, Giudice, Moratz, & Worboys, 2011; Misra, Ikbal, Bourlard, & Hermansky, 2004; Toh, Togneri, & Nordholm, 2005). We find that previous specifications do not match our filterbank and window decomposition specifications.

To interpret this feature in the frequency domain, we first need to transform our STFT spectrum into a probability mass function (PMF). When the PMF is maximally flat, the entropy is high, corresponding to a state of maximum uncertainty or 'disorder'. In contrast, when the PMF has one sharp peak, it corresponds to a state of low uncertainty, where the entropy is said to be low. To convert our spectrum to a PMF $p(x_j)$, we divide the frequency constituents of the power spectrum X_j by the sum of all frequency constituents of the same spectrum, defined as follows:

$$p(x_j) = \frac{X_j}{\sum_{j=1}^{N} X_j}$$

where X_j is the power of $j = 1 \dots N$ frequency constituents. We repeat this procedure for each sub-band, necessarily resulting in 10 PMFs. For each PMF $p(x_j)$ we compute the Shannon Entropy:

$$H(X) \coloneqq -\sum_{j=1}^{N} p(x_j) \cdot log_2 p(x_j)$$

The resulting feature is the Sub-Band Entropy, consisting of 10 spectral entropy sub-band vectors.

3.3.2 Sub-Band Skewness

Spectral skewness is the third central moment of an STFT's probability density function (PDF). Skewness is a measure of symmetry: when spectral skewness has a positive value, the distribution is positively skewed to the right containing larger values than the mean. A symmetrical distribution has a skewness value of zero. We obtain the coefficient of skewness from the following expression:

$$Skewness_{coef} = \frac{E(x-\mu)^3}{\sigma^3}$$

Where $E(x)$ is the expected value of x with x being the data observed of which μ is the mean and σ the standard deviation. By repeating the computation for each sub-band, the resulting feature is sub-band skewness, consisting of 10 spectral skewness sub-bands. In the literature we find two relevant applications (Seo & Lee, 2011; Yeh, Roebel, & Rodet, 2010), of which one (Seo & Lee, 2011) had a similar approach for GTZAN. Despite the similar approach both studies do not match our filterbank and window decomposition specifications.

3.3.3 Sub-Band Kurtosis

Spectral Kurtosis refers to the fourth central moment of an STFT's probability density function (PDF). Kurtosis indicates whether a PDF is flat or peaky near its mean value, it is a measure of the peakedness of the distribution. As seen below, the Kurtosis coefficient is given by dividing the fourth cumulant by the square of the variance of the distribution:

$$Kurtosis_{coef} = \frac{E(x-\mu)^4}{\sigma^4} - 3$$

Where $E(x)$ is the expected value of x with x being the data observed of which μ is the mean and σ the standard deviation. A normal distribution has Kurtosis = 3, for this reason the " - 3'' constant is used to balance out the kurtosis value of the normal distribution. The kurtosis coefficient is obtained for each sub-band resulting in the sub-band kurtosis feature. In literature we find one relevant work (Seo & Lee, 2011) and two distantly relevant works (Sällberg, Grbić, & Claesson, 2007; Yermeche, Grbic, & Claesson, 2007). In every work, the entire extraction specification (filterbank, window decomposition, kurtosis implementation) do not match our own.

3.3.4 Sub-Band Zero Crossing Rate

The zero-crossing rate (ZCR) is used extensively in the fields of speech recognition and music information retrieval. The idea behind ZCR is to compute the average rate of sign changes for a signal in the time domain. Essentially, we count a crossing or sign change when the signal crosses to the positive or the negative range of values. For a signal window $x(n)$, we calculate the zero-crossing rate as follows:

$$ZCR \triangleq \frac{1}{2} \cdot \sum_{n=2}^{N} \left| sign\big(x(n)\big) - sign(x(n-1)) \right|$$

where,

$$sign(x) = \begin{cases} 1, if\ x > 0 \\ 0, if\ x = 0 \\ -1, if\ x < 0 \end{cases}$$

The computation occurs for every sub-band, resulting in 10 sub-band zero crossing rates vectors.

3.3.5 Sub-Band Flux

Sub-band flux was first introduced by Alluri and Toiviainen (2010) and is the only perceptually validated feature in our study. In the past, sub-band flux has had several uses in: 1) Timbre research (Alluri, 2012; Alluri & Toiviainen, 2009, 2010, 2012; Alluri et al., 2012; Eerola, Ferrer, & Alluri, 2012); 2) Music and movement research (Burger, 2013); 3) Music genre classification (M. A. Hartmann, 2011; M. Hartmann, Saari, Toiviainen, & Lartillot, 2013); 4) Music and neuroscience research (Alluri, 2012; Alluri et al., 2012; Hoefle et al., 2018).

Sub-band flux is based on the spectral flux feature as used in a plethora of studies and applications. The 'flux' part of the feature is a measure of a signal's temporal fluctuation, as a function of the distance between two successive windows. In the spectral case, instead of the raw signal, it is computed for the STFT spectrum resulting in spectral flux. Analogically, spectral flux measures the temporal fluctuation of the magnitude spectra between two successive windows. Bellow, we see the Euclidian distance metric used in our implementation:

$$d = \sqrt{\sum_{n=1}^{N} (x_t[n] - x_{t-1}[n])^2}$$

Where at times t and $t-1$ the two windows are normalized to have the Euclidean norm:

$$\Sigma x[n]^2 = 1$$

The computation occurs in every sub-band, resulting in the sub-band flux feature.

3.3.6 Mel-frequency Cepstral Coefficients (MFCCs)

The Mel-frequency-cepstral coefficients (Logan, 2000; Mermelstein, 1976) are computed in a cascade of five steps shown in figure 14; These steps are: 1) Window a signal into overlapping windows across its temporal length; 2) The STFT of a signal is computed for each window; 3) Each power spectrum is filtered with mel-frequency spaced triangular filters

allowing for the perceptual positioning of the filters in the frequency domain; 4) The energies of every triangular filter are summed, and the logarithm of the energies is taken; 5) The discrete cosine transform (DCT) is applied to the logarithmic energies. The resulting features are the MFCC coefficients, typically a portion of the coefficients is maintained. In this study, we extracted thirteen coefficients, starting from the first coefficient (zero is discarded). Each input signal was decomposed into 25 millisecond windows with a 50% overlapping/hop size prior to the MFCC computation.

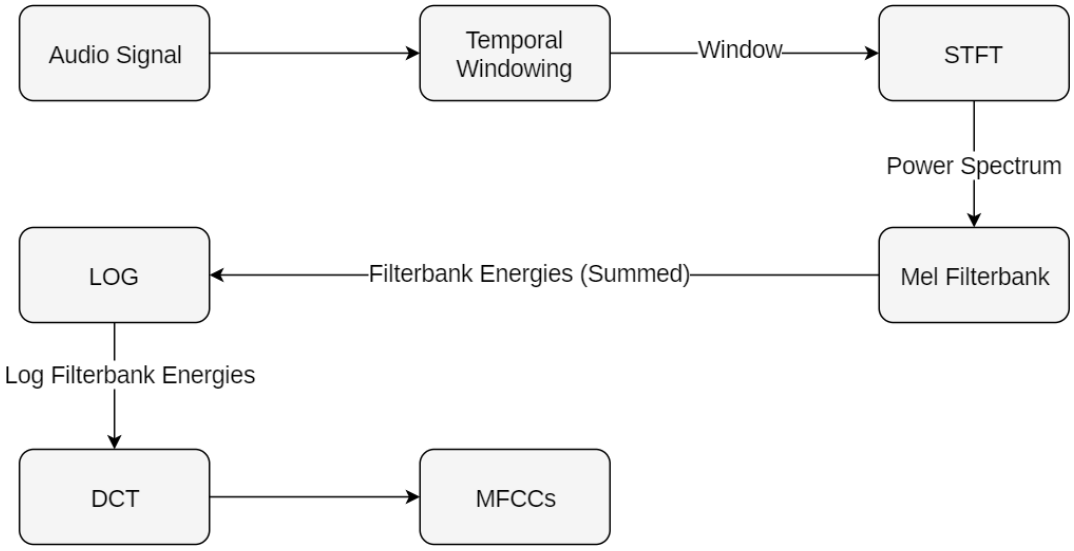

FIGURE 14. The MFCC extraction pipeline, each line indicates the output of each operation.

3.3.7 Feature Statistical Summarization

The final step of our feature extraction is the statistical summarization of the features. Each feature consisted of ten vectors/sub-band components extracted with a different window size via FDW. To reduce classifier learning time and compactify each vector, we summarized each sub-band feature component with its mean and standard deviation values. This procedure produced two statistic values per sub-band feature component.

3.4 Feature Selection & Combinatorial Sub-Sets

In this last section of the pre-processing stage, we generated a multitude of feature sub-sets which became the inputs to the classification stage. We created feature subsets in three ways, manually, semi-manually and algorithmically. In this section, we elaborate on each feature selection strategy. The entire feature selection and sub-set generation pipeline is shown in figure 15 at the end of the section.

3.4.1 Manual Selection

In the manual selection case, we devised two feature sub-sets and further generated more sub-sets between statistical summaries; When selecting feature mean values we code-name the sub-set as 'Feature sub-set (μ)', when we select the standard deviation values we used the code-name 'Feature sub-set (σ)', when both summaries were used, we aggregated both code-names as 'Feature sub-set (μ, σ)'. In tables 15 and 16 we show each manually selected feature sub-set, dimensionality and description.

TABLE 15. The 'All Features' sets, comprising of all features and every summary statistic sub-set.

'All Features' Sets	**Dimensionality**	**Description**
'All Features $(\boldsymbol{\mu})$'	63	Only mean values of all features.
'All Features $(\boldsymbol{\sigma})$'	63	Only standard deviation values of all features.
'All Features $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	126	Mean and standard deviation values of all features.

TABLE 16. The 'individual feature' sets, comprising of each individual feature and both summary statistics.

'Individual feature' Sets	**Dimensionality**	**Description**
'SB-Entropy $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	20	Mean and standard deviation values of 10 sub-band entropies.
'SB-Flux $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	20	Mean and standard deviation values of 10 sub-band fluxes.
'SB-Kurtosis $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	20	Mean and standard deviation values of 10 sub-band kurtosis.
'SB-ZCR $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	20	Mean and standard deviation values of 10 Sub-Band zero crossing rates.
'SB-Skewness $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	20	Mean and standard deviation values of 10 sub-band skewness.
'MFCCs $(\boldsymbol{\mu}, \boldsymbol{\sigma})$'	26	Mean and standard deviation values of 13 MFCC coefficients.

3.4.2 Semi - Manual Selection

In semi-manual selection, we devise the 'Top 2' feature sub-set from the classification performance ranking of the 'individual features' sets. This selection design deals with two features only, for this reason, the μ_1, σ_1 codes referred to the mean and standard deviation of the best performing individual feature, while μ_2, σ_2 referred to the same statistics for the second-best feature. Table 17 displays the composition of our 'Top 2' semi- manual selection design.

TABLE 17. The semi-manual feature selection subsets, containing the top two performing individual features.

Top 2 Feature Sub-Set	**Dimensionality** (with or without MFCCs selected)	**Description**
'Top 2 [Feature 1 $(\boldsymbol{\mu_1}, \boldsymbol{\sigma_1})$ – Feature 2$(\boldsymbol{\mu_2}, \boldsymbol{\sigma_2})$]'	46 or 40	Mean and standard deviation values of the top performing feature with that of the second-best feature.
'Top 2 [Feature 1 $(\boldsymbol{\mu_1})$ – Feature 2 $(\boldsymbol{\sigma_2})$]'	23 or 20	Mean values of the first feature with standard deviation values of the second feature.
'Top 2 [Feature 1 $(\boldsymbol{\sigma_1})$ – Feature 2 $(\boldsymbol{\mu_2})$]'	23 or 20	Mean values of the second feature with standard deviation values of first feature

3.4.3 Algorithmic Feature Selection

Algorithmic feature selection or simply 'feature selection' plays an important role in automatically ranking and selecting relevant features for a classification task. Often a portion of features involved in machine learning may not be informative or relevant to the classification task. Indeed, a high number of irrelevant features may increase the complexity of a model and even decrease classification and computational performance by increasing overfitting or imposing other unwanted effects. Feature selection algorithms help to combat such effects. Importantly, feature selection should not be perplexed with dimensionality reduction methods, both reduce the number of features for a given task, but dimensionality reduction methods may produce different features from the initial feature set, feature selection methods do not.

There are three types of feature selection methods, filter methods, wrapper methods and embedded methods. Filter methods employ statistical measures to score each feature with

respect to the dependent variable or independently. In contrast, wrapper methods generate feature combinations and compare the classification accuracy results of such combinations directly via the classification stage. Embedded methods identify feature contributions to classification accuracy within and during the classification process. A popular class of embedded methods is regularization techniques often used to penalize classification and regression algorithms such that they may reduce over-reliance on specific features, this often may lead to reduced overfitting.

The choice of feature selection algorithm depends heavily on the understanding of a given problem. There is no 'one size fits all' selection algorithm, often selection algorithms tend to produce completely different rankings for the same input features. In the absence of deep problem understanding it is common for multiple selection methods to be evaluated, and at times aggregated ranks between multiple rankings may be performed. In our study, we employ a filter method using the Information gain (IG) algorithm that we detailed below. The resulting feature sub-set was code-named 'Information Gain Top 20' to refer to the top 20 features suggested by IG when inputting the 'All Features (μ, σ)' feature set. We implemented IG in python with the Orange3 library (Demšsar et al., 2013).

3.4.4 Information Gain

Information Gain (IG) is a filter method computed for each feature with respect to the class labels (Liu & Motoda, 1998); it relies heavily on the Shannon's information entropy (Shannon, 2001). To obtain $IG(X|Y)$ were X and Y are random variables, let us consider the formula for the information entropy H of variable X:

$$H(X) = -\sum_{x \in X} p(x)\, log_2\big(p(x)\big)$$

Where $p(x)$ is the marginal probability density function for X. When we introduce variable Y and devise the values of variable X with respect to the values of Y, a relationship between X and Y exists only when the entropy of X devised by Y is smaller than the initial entropy of X, the entropy of X devised by Y is given by:

$$H(X|Y) = -\sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) log_2\big(p(x|y)\big)$$

Where $p(y)$ is the marginal probability density function for variable Y and $p(x|y)$ is the conditional probability of x given y. After the conditional entropy step, information gain is defined as the amount of entropy decrease in X represented by the surplus information provided from Y about X, formally defined as:

$$IG(X|Y) = H(X) - H(X|Y)$$

Importantly, $IG(X|Y) = IG(Y|X)$ because information gain is symmetrical for the two variables (Yu & Liu, 2003).

3.4.5 Feature Selection Overview

To summarize, figure 15 maps the flow of our feature selection sets, along with their statistical summary subsets with respect to the classification stage.

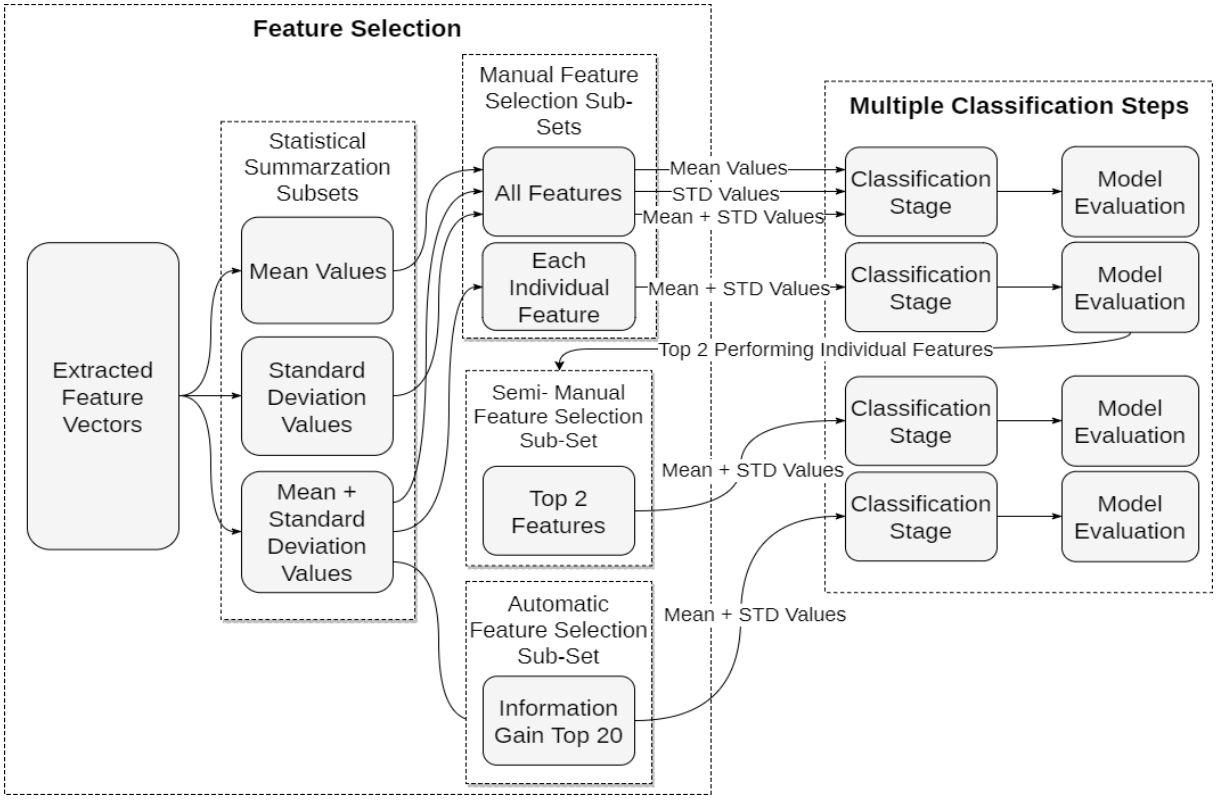

FIGURE 15. The flow of feature selection subsets with respect to the classification stage.

3.5 The Classification Stage

In the classification stage, our learning tasks (music genre, music mood) are evaluated by performing supervised machine learning with three learning algorithms. The input to each algorithm consists of our feature selection sets. In this section, we detail each learning algorithm and evaluation criteria used in this study.

3.5.1 Learning Algorithms & Evaluation

In order to perform supervised machine learning, we employ three learning algorithms: Support Vector Machines (SVM), Multinomial Logistic Regression (MLR) and K-Nearest Neighbors (K-NN). The choice of SVM stemmed directly from the literature as the majority classifier of top MIREX entries. The choice of MLR and K-NN provides diversity in the learning approach because fundamentally all three algorithms operate and learn under different principles.

Whenever each learning algorithm is trained on the data it produces a final classifier model. Each model is evaluated by predicting data examples excluded from the training process (the testing data). To train and evaluate with 'unknown' data from a single dataset, our data are split into training and testing sets as dictated by 10 - fold stratified cross-validation. Every classification task was implemented in Python with the Scikit-learn library (Pedregosa et al., 2011).

3.5.2 Stratified Cross-Validation

In multiclass classification, often some classes have more observations than others and vice versa as it is the case for both our datasets. Because of this, we employed a stratified cross-validation scheme such that we could maintain the distribution of class observations in the training/testing sets of each fold. In this fashion, the training sets and the testing sets will have a proportional number of examples from each class according to the original data to class distributions. This procedure lowers the probability of invalid classification results (Saari, 2009).

3.5.3 Artist Filter Cross-Validation

In the case where artist filtering is used we employ two-fold non-stratified cross-validation, following suit to other AF GTZAN implementations (Sturm, 2013b, 2014b).

3.5.4 Fold Standardization and Scaling

We retained a typical standardization procedure for each iteration in our cross-validation scheme. Bellow we see the z value calculation:

$$z = \frac{x - \mu}{\sigma}$$

We first began with each training set, where the mean value μ of an entire feature gets subtracted for each of its observations x. Subsequently, we divide the feature observations by that feature's standard deviation σ. The output z is a feature with a mean, $\mu = 0$ and standard deviation, $\sigma = 1$. As a final step, we apply the standardization parameters of the training set to the corresponding testing set and repeat the entire process for each cross-validation iteration. In this way, we ensure that the testing sets in each iteration are scaled appropriately to their corresponding training sets (Saari, 2009).

3.5.5 Performance Metric

In this study, we employ the average classification accuracy (CA) across folds as our figure of merit, we refer to the average CA, only as 'Accuracy'.

3.6 Classification Algorithms

3.6.1 Support Vector Machines

Support vector machines (Boser, Guyon, & Vapnik, 1992), is one of the most widely used algorithms in supervised learning. The algorithm is known for good performance, robustness, flexibility and computational efficiency (Baesens et al., 2003; Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, & Bernhard Schölkopf, 2006; Van Gestel et al., 2004). To understand the idea behind support vector machines, let us consider a two-class problem in n

dimensions. In a linear separation case, we are looking for the hyperplane with dimensionality $n-1$ that best separates the two classes in our feature space. For a nonlinear separation case, we use nonlinear kernels and look for the best separating hyperplane within a transformed higher dimensional space. A new observation is classified with respect to its position relative to this hyperplane.

In more detail (Vapnik, 2013), let us consider a set of data *N* points $\{(x_{j,}y_{j,})\}_{j=1}^{N}$, from this set, our features are $x_j \in \mathbb{R}^n$ with the corresponding binary ground truth labels $y_j \in \{(-1,+1)\}$, the SVM conditions that are satisfied are:

$$\begin{cases} w^T\phi(x_j) + b \geq +1, & if\ y_j = +1 \\ w^T\phi(x_j) + b \leq -1, & if\ y_j = -1 \end{cases}$$

The expression is equivalent to $y_j[\, w^T\phi(x_j) + b] \geq 1,\ j = 1, \ldots, N$. Figure 15a can help to visualize on an adapted margin optimization problem (Martens, Baesens, & Gestel, 2009).

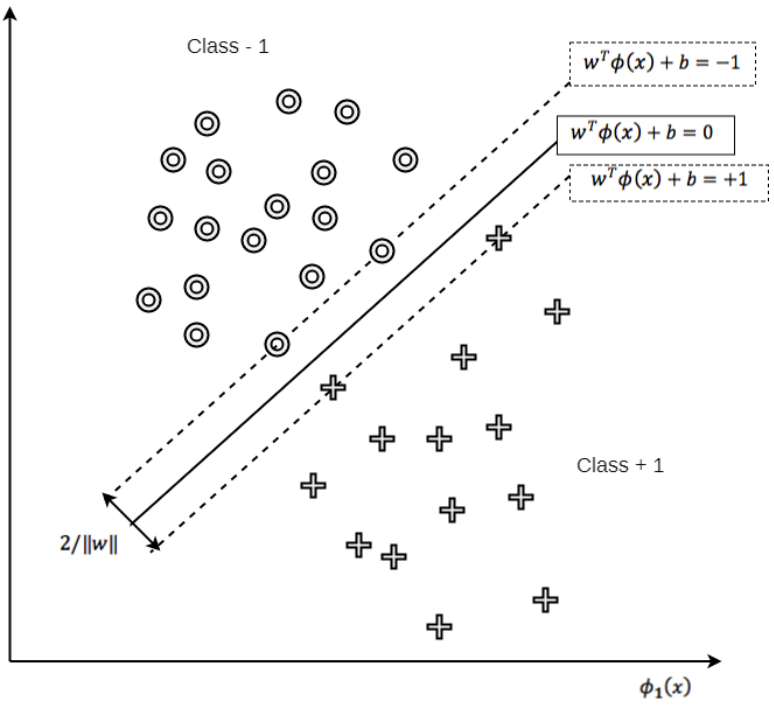

FIGURE 15a. Adapted SVM margin optimization problem (Martens et al., 2009).

The nonlinear function $\phi(x_j)$ maps the input features into a higher dimensional feature space, and b corresponds to the adjustable bias. As a consequence of the introduced inequalities another hyperplane $w^T\phi(x_j) + b = 0$ separates the two classes. The objective of the classifier is to maximize the margin between both classes by minimizing $w^T w$.

Next, the classifier is formulated in primal weight space as:

$$y(x) = sign\,[w^T\phi(x) + b]$$

Conclusively, the problem is defined as a convex optimization problem and optimized utilizing the Lagrangian where we obtain the following classifier from the solution:

$$y(x) = sign\left[\sum_{j=1}^{N} a_j y_j K(x_j, x) + b\right]$$

Where $K(x_j, x) = \phi(x_j)^T\phi(x)$, which is the kernel function that satisfies the Mercer theorem (Mercer, 1909) and a_j are the Lagrange multipliers computed from the following optimization problem:

$$\max a_j\ -\frac{1}{2}\sum_{j,i=1}^{N} y_j\, y_i K(x_j\,, x_i) a_j a_i + \sum_{j,i=1}^{N} a_j$$

Subject to constrain:

$$\begin{cases} \sum_{j=1}^{N} a_j y_j = 0 \\ 0 \le a_j \le C, j = 1, \dots, N \end{cases}$$

With $C \in \mathbb{R}_{+,}$ where C is an adjustable parameter and the problem becomes a Quadratic programming problem in a_j.

The choice of a kernel function can vary depending on the problem, bellow we find some popular options with T, j, d, σ being constants:

- Linear Kernel: $K(x_j, x) = x_j^T x$
- Polynomial Kernel: $K(x_j, x) = (1 + x_j^T x/c)^d$
- Radial Basis Function Kernel: $K(x_j, x) = exp\{-\|x - x_j\|_2^2/\sigma^2\}$

Where T stands for transpose, j for index and d determining polynomial degree. In this study we used the radial basis function (RBF) and it's default hyper parameter values as set in the Scikit-learn library (Pedregosa et al., 2011). The values used were, $C = 1$ and $\gamma = \frac{1}{total\ number\ of\ feature\ dimensions}$ referred to as 'auto' within the library.

3.6.2 Logistic Regression

Logistic regression employs the logistic function and fits the data to the logistic curve; this allows the model to predict a binary outcome for the fitted data (Hosmer Jr, Lemeshow, & Sturdivant, 2013). The term 'logistic' derives from the use of the logistic curve shown in figure 16, logistic regression belongs to the class of generalized linear models. Below, we find the formulation of the logistic function:

$$f(q) = \frac{e^q}{e^q + 1} = \frac{1}{1 + e^{-q}}$$

Where:

$$q = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \dots \beta_j x_j$$

Except for β_0 which is a constant, $\beta_1, \beta_2, \beta_3 \dots \beta_j$ would correspond to feature values in the form of regression coefficients; these are the coefficients we want to learn from our data. Due to a value range from 0 up to 1 for the logistic curve, the fitted data may be interpreted in terms of probabilities.

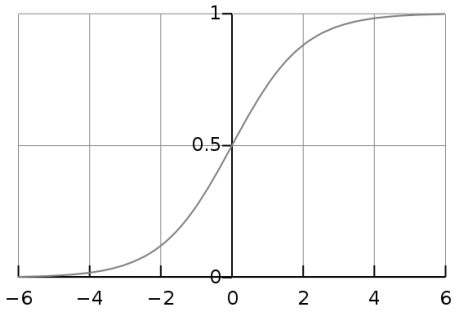

FIGURE 16. The logistic function plotted in range.

Logistic regression requires a binary class, (i.e. The response variable must be binary). In our study, the response variables have more than two classes; thus, we employ the multinomial logistic regression (MLR) case. MLR allows for multiclass classification (Böhning, 1992; Krishnapuram, Carin, Figueiredo, & Hartemink, 1992), where the set-up is almost identical to logistic regression with the difference that our response variables are categorical and have k possible outcomes (classes). Consequentially q expands to the linear prediction function $q(k, i)$ where we predict the probability that an observation i has the class (outcome) k, such that:

$$q(k, i) = \beta_{0,k} + \beta_{1,k}x_{1,i} + \beta_{2,k}x_{2,i} + \beta_{3,k}x_{3,i} + \beta_{4,k}x_{4,i} \dots \beta_{j,k}x_{j,i}$$

Where $\beta_{j,k}$ is our regression coefficient with the jth feature (explanatory variable) and kth class (outcome). To reduce overfitting effects, we also use L2 Regularization which penalizes a portion of the model weights if they increase substantially.

3.6.3 K-Nearest Neighbors

The K-nearest neighbors (K-NN) algorithm is one of the most ‘straightforward’ algorithms to implement, frequently used in both classification and regression problems (Dudani, 1976). It is a non-parametric method, meaning that it can approximate irregular decision boundaries. The classifier is also an ‘instance based’ learner or ‘lazy’ learner, it stores the entire training set in memory and only builds a model when the testing data is evaluated. K-NN is unlike other algorithms such as SVM, where the model is already constructed from the training data.

The algorithm operates under the assumption that data that are close to each other, may be similar. It deals with distances between data points with a given distance function d, where, given a new observation x_{new} the algorithm looks at the K instances from the training data with the least mutual distance from the new observation x_{new}. The algorithm assigns a class for x_{new}, based on the majority class of its K-neighbours, referred to as 'majority voting'. Given x_{new} the class assignment is calculated as the posterior probabilities for each K neighbour without assuming any probability distribution for the testing data. We use the algorithm with the default Scikit-learn implementation ($K = 5$, uniform weights and Minkowski distance).

3.7 Experimental Design Flowchart

We dedicate this final page to figure 17 which maps the flow of all operations in our experimental design.

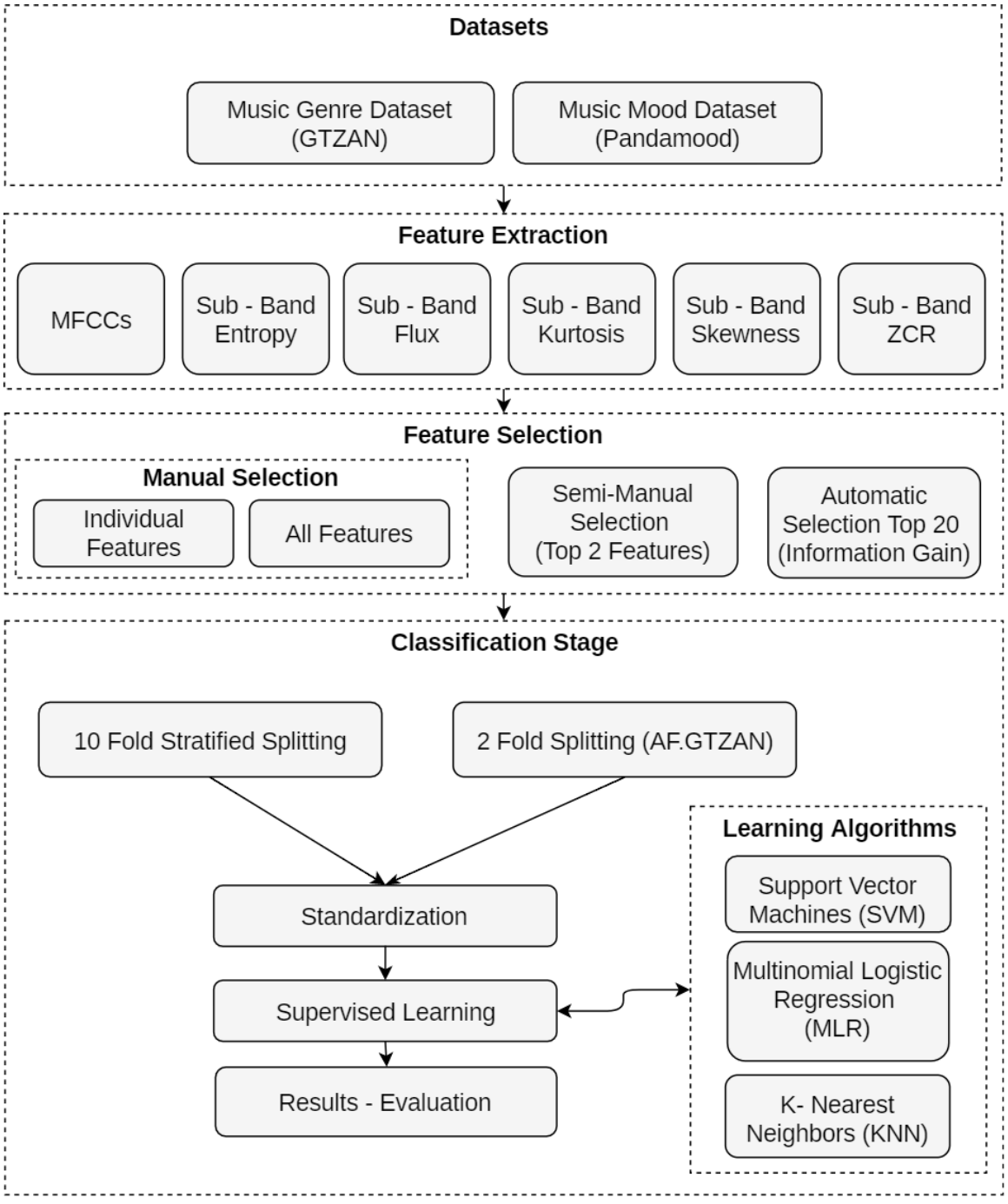

FIGURE 17. The flow of operations in the experimental design.

4 RESULTS

In this chapter, we examine the outcomes of our classification experiments. Our primary goal in this study was to evaluate the classification performance of the FDW sub-band features against the MFCCs in music genre and music mood classification tasks. Our reporting set up is such that it mirrors and expands that found in MIREX and other relevant works. We used ten-fold standardized and stratified cross-validation for both tasks, excluding GTZAN artist filtering which we evaluated with two-fold unstratified cross-validation.

In Appendix A we list all the classification results from the 117 models that we evaluated. Due to the shire size of our experimental design this chapter's figures focus only on the highest performing classifier within each task. We found that multinomial logistic regression performed the highest in GTZAN and support vector machines in PandaMood. We exclusively use box-plots that show the mean (dotted line), median (solid line) and standard deviation (dotted line) of the classification accuracies we obtained with cross-validation.

We begin with the results section of each classification task separately, starting with GTZAN. Every task has the following feature set reporting order: 1) 'All features' and their statistical summary sets; 2) Individual feature sets; 3) Semi-manual and algorithmic feature selection sets. After reporting all feature selection sets for every task, we construct and analyze the 'top 5 models' as a result of rank aggregation. The chapter concludes with two rankings and an analysis of task specific feature importance as obtained from automatic (information gain) and manual feature selection (individual features set) approaches.

4.1.1 GTZAN Results

In figure 18 we see the classification accuracies of the 'All Features' sets, we observe that testing was consistently higher than artist filter testing. When considering each feature set, 'All Features (μ, σ)' had the highest average testing score (77.83%), standard deviation (4.57%) and artist filtered score (64.57%). Despite the high testing scores, the training to testing distance was 21.7% which indicates a high chance of overfitting. This finding was not surprising given that this feature set had the highest dimensionality. When ranking all testing scores in descending order, we obtain the following: 1) 'All Features (μ, σ)'; 2) 'All Features (μ)'; 3) 'All Features (σ)'. This ranking order shifts for training to testing distances (ascending order): 1) 'All Features (μ, σ)'; 2) 'All Features (σ)'; 3) 'All Features (μ)'. Importantly, the testing ranks do not match the artist filter testing ranks, which were (descending order): 1) 'All Features (μ, σ)'; 2) 'All Features (σ)'; 3) 'All Features (μ)'. The ranking discrepancy between artist filtered and un-filtered models is hard to explain as it unclear if these inconsistencies occurred due to the varying feature sets or the artist filter itself.

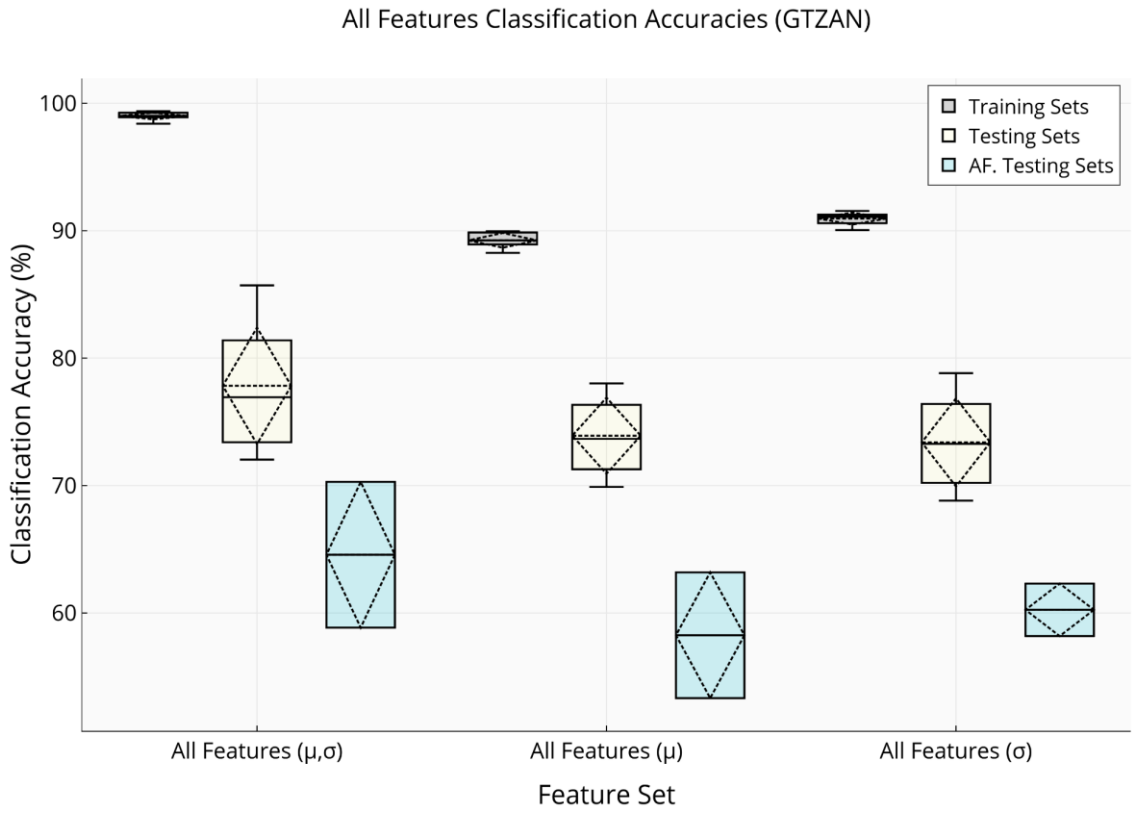

FIGURE 18. Classification accuracies of GTZAN 'All Features' statistical summary sets, each box plot displays the mean, standard deviation (dotted line) and median (solid line).

Figure 19 shows all individual feature set classification accuracies. Similarly to figure 18, we see that artist filtered models performed lower than the non-filtered models. Between all individual features, we find that 'SB- Entropy (μ, σ)' had the highest average accuracy in testing (66.77%), artist filter testing (53.05%) and the lowest absolute training to testing distance (6.05%). We find outliers in the training and testing sets of SB-Kurtosis and SB-ZCR. When ranking for testing accuracy we obtain the following (descending order): 1) SB-Entropy (66.77%); 2) SB-Flux (63.99%); 3) SB- Kurtosis (61.81%); 4) SB-Skewness (59.90%); 5) MFCCs (58.57%); 6) SB-ZCR (56.00%). The testing ranks for artist filtering are identical to the non-filtered ranks except for '2) SB-Flux' and '3) SB-Kurtosis', that swap places. These ranking contradictions follow from our findings in figure 18 and further suggest a somewhat independent classifier behavior with artist filtering. In perspective, we see that the sub-band features, except SB-ZCR, outperformed the MFCCs both in testing accuracy and AF.Testing accuracy. Moreover, the sub-band features had a lower training to testing distance than the MFCCs, with an average of 6.61% (SB-Features) against 9.03% (MFCCs).

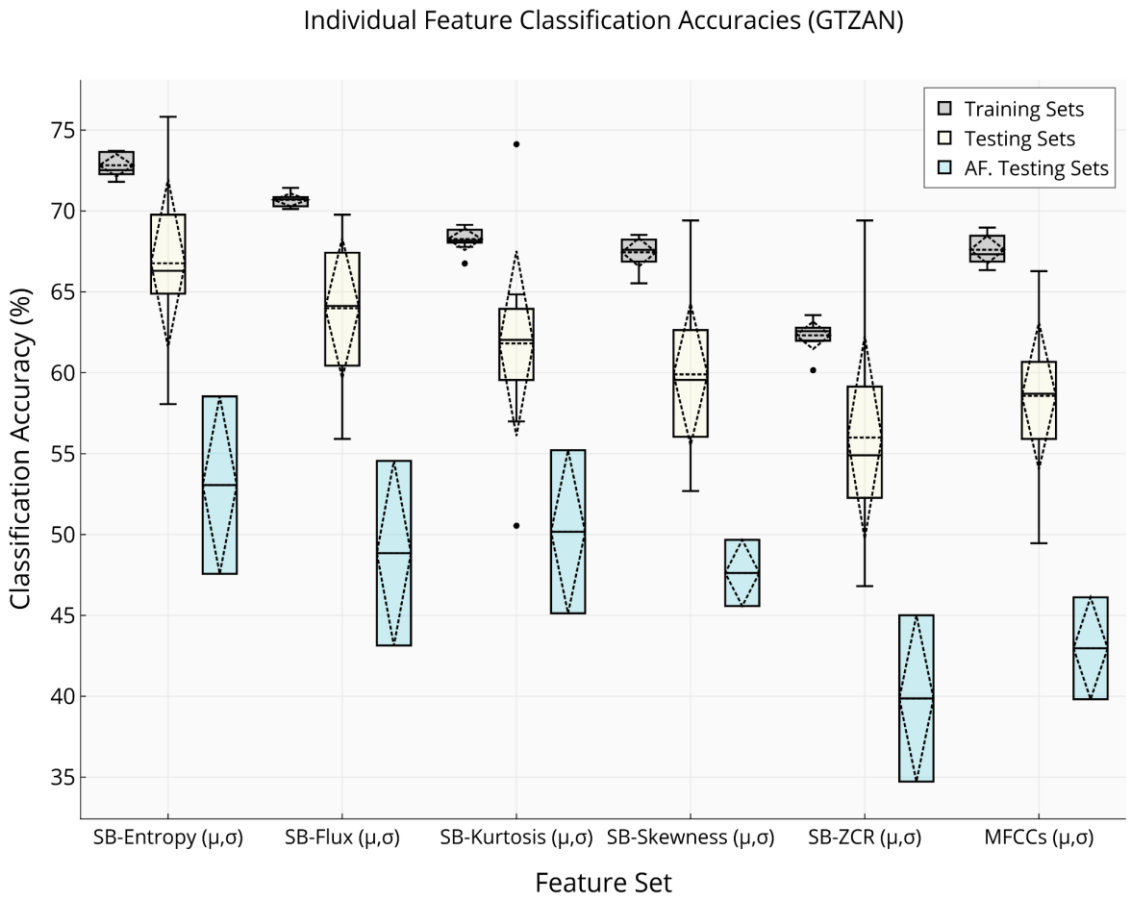

FIGURE 19. Classification Accuracy of GTZAN Individual features, each box plot displays the mean, standard deviation (dotted line) and median (solid line).

In figure 20 we see the classification accuracies of all feature selection sets. Similarly to figures 18 and 19, artist filtered models underperformed against the unfiltered models. At this point we clearly see that artist filtering consistently decreased classification accuracy for GTZAN. On the feature set level we see that 'SB- Entropy (μ, σ) & SB- Flux (μ, σ)' performed the best in testing (74.02%) and AF.testing (58.81%), along with an 11.20% in absolute training to testing distance. This result surprised us since we did not expect semi-manual selection (top 2 features) to outperform automatic feature selection. Furthermore, IG ranked second in testing accuracy, but the remaining feature sets were outperformed with an average difference of only 0.45%. In contrast, SB- Entropy (μ, σ) & SB- Flux (μ, σ) outperformed all other feature sets with a distance larger than 8%. To encapsulate the testing score order the following ranking was obtained (descending order): 1) SB- Entropy (μ, σ) & SB- Flux (μ, σ); 2) Information Gain (Top 20); 3) SB-Entropy(μ) & SB-Flux(σ); 4) SB-Entropy(σ) & SB-Flux(μ). In addition, when we consider the AF.testing ranks we obtain: 1) SB- Entropy (μ, σ) & SB- Flux (μ, σ); 2) SB- Entropy (μ) & SB- Flux (σ); 3) SB-Entropy (σ) & SB- Flux (μ); 4) Information Gain. As in figures 18 and 19 the filtered ranks did not match the unfiltered ranks, which further suggests some potential degree of independence between the two.

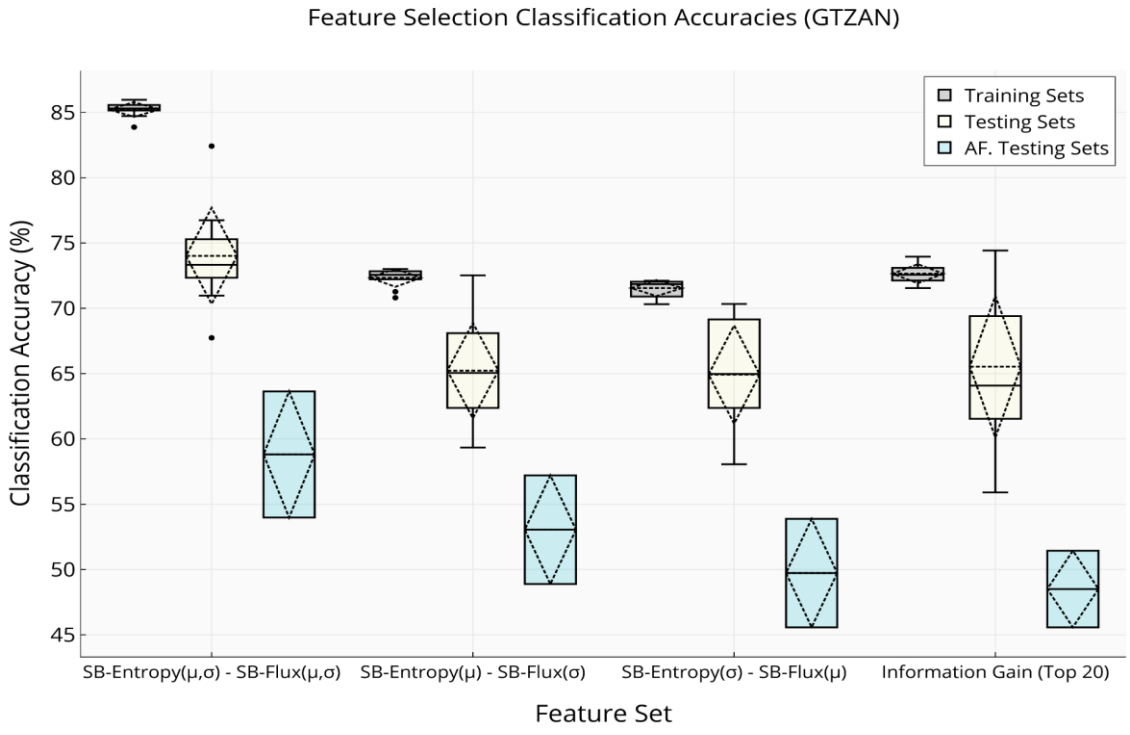

FIGURE 20. Classification accuracy of GTZAN feature selection sets, each box plot displays the mean, standard deviation (dotted line) and median (solid line).

4.1.2 PandaMood Results

In figure 21 we see the classification accuracies of the 'All Features' sets. We can observe that the 'All Features (μ, σ)' set had the highest average accuracy (42.41%), the highest standard deviation (5.15%), most outliers and the highest training to testing distance (27.79%). The high training to testing distance was strongly indicative of overfitting, which was further manifest in both 'All Features (μ)' and 'All Features (σ)' each with a gap larger than 25%. Regarding testing accuracies, the following ranking was obtained (descending order): 1) 'All Features (μ, σ)'; 2) 'All Features (σ)'; 3) 'All Features (μ)'. Inversely, the ranking for training to testing distances was (ascending order): 1) 'All Features (μ)'; 2) 'All Features (σ)'; 3) 'All Features (μ, σ)'. For both rankings, mean rank aggregation would be inconclusive as it would simply result in the same ranking position for all feature sets.

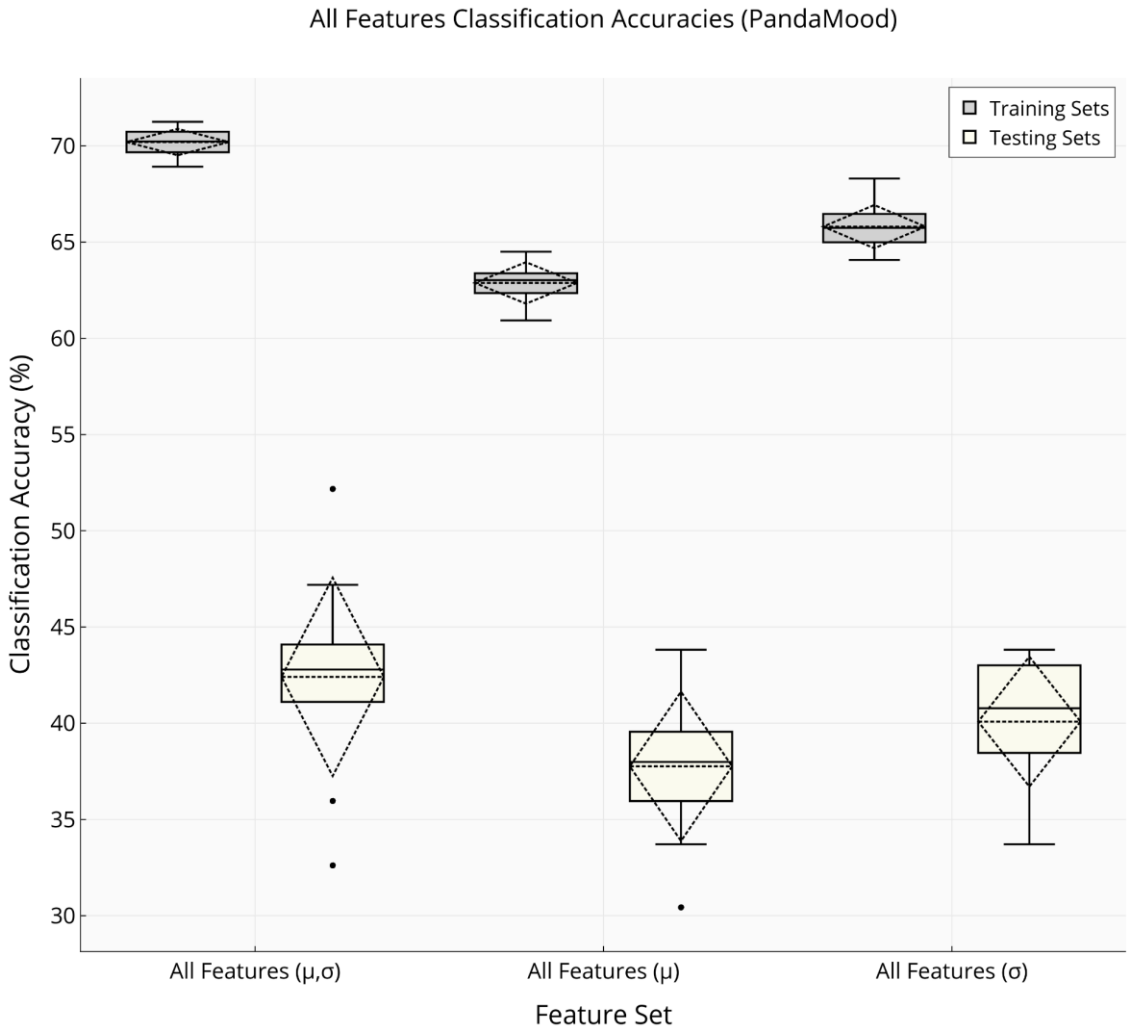

FIGURE 21. Classification accuracies of PandaMood 'All Features' statistical summary sets, each box plot displays the mean, standard deviation (dotted line) and median (solid line).

In figure 22 we see all 'individual feature' set classification accuracies. We discern that SB-Entropy (μ, σ) had the highest average testing score (39.51%), most outliers, the fifth highest standard deviation (4.19%) and the second largest training to testing distance (20.65%). In perspective, this kind of classifier behavior is indicative of overfitting. When ranking all average testing scores in descending order, we obtained the following: 1) SB-Entropy (39.51%) ; 2) MFCCs (38.65%); 3) SB-Skewness (37.52%); 4) SB-Flux (36.29%) 5) SB-Kurtosis (36.06%); 6) SB-ZCR (35.40%). The highest standard deviation is found in SB-Skewness (5.41%) and the lowest in SB-Flux (2.91%). The average testing score distance from the last and the first rank item was only 4.11%. The ascending ranking of mean training to testing distances was: 1) SB-Kurtosis (12.73%); 2) SB-Skewness (14.91%) ; 3) SB-Flux (17.47%); 4) SB-ZCR (18.79%) ; 5) SB-Entropy (20.65%); 6) MFCCs (26.59%). The range between the first and last item was wider (13.86%) than the range found in testing accuracies (4.11%), this means that there is a wider range in overfitting indicators.

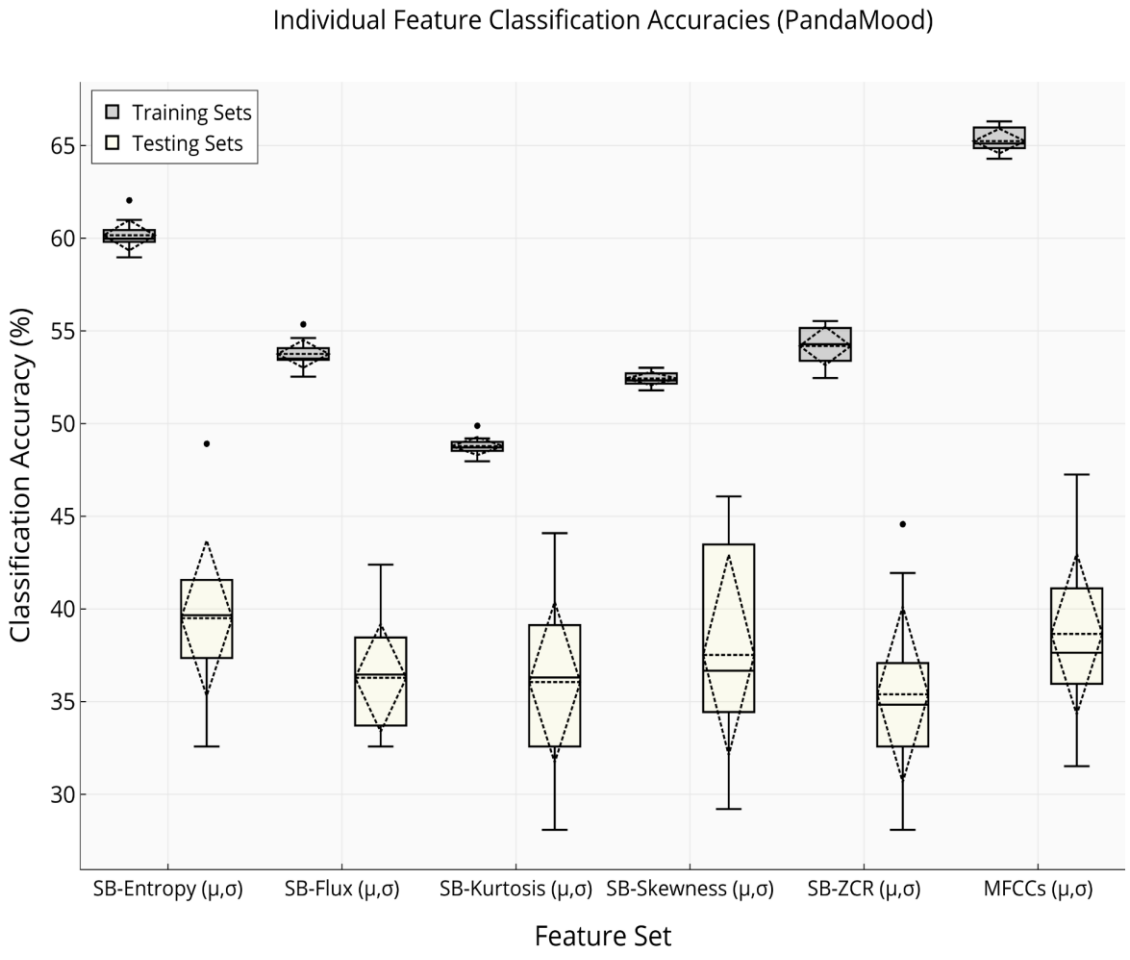

FIGURE 22. PandaMood classification accuracy of individual feature sets, each box plot displays the mean, standard deviation (dotted line) and median (solid line).

In figure 23 we see the classification accuracies of semi-manual and automatic feature selection. We observe that SB-Entropy(μ, σ) & MFCCs (μ, σ) has the highest average testing score (42.18%), the largest training to testing distance (27.88%), and the lowest standard deviation (2.89%). A 27.88% training to testing distance is highly suggestive of overfitting effects. Outliers were found in the training and testing set of Information Gain, while the largest standard deviation (5.42%) was found in SB-Entropy(μ) & MFCCs (σ). With respect to all average testing scores, we obtain the following ranking (in descending order): 1) SB-Entropy(μ, σ) & MFCCs (μ, σ); 2) SB-Entropy(μ) & MFCCs (σ); 3) SB-Entropy(σ) & MFCCs (μ); 4) Information Gain (Top 20). Interestingly, the rankings shifts for training to testing distances (in ascending order); 1) Information Gain (Top 20); 2) SB-Entropy(μ) & MFCCs (σ); 3) SB-Entropy(μ, σ) & MFCCs (μ, σ); 4) SB-Entropy(σ) & MFCCs (μ); To our surprise, we see that semi-manual selection outperformed automatic selection, yet automatic selection had the smallest training to testing distance.

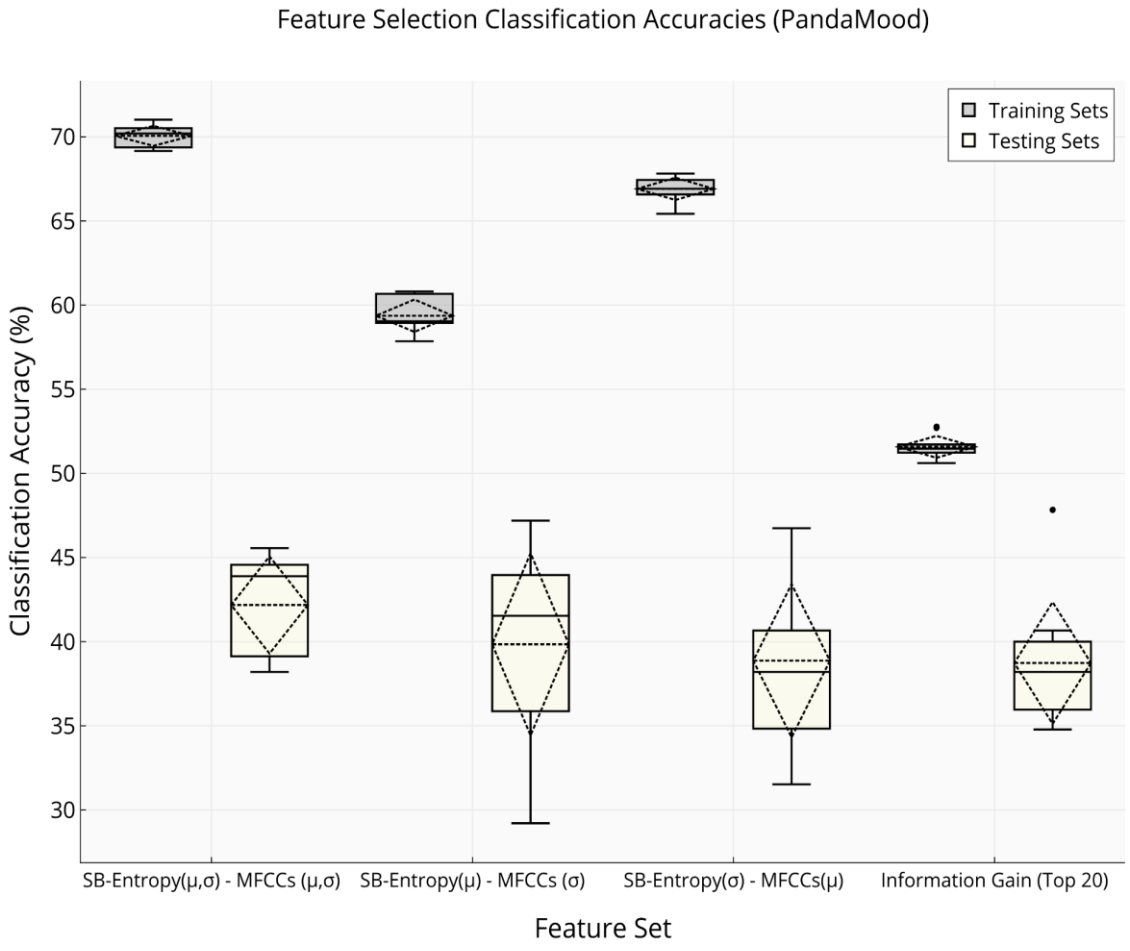

FIGURE 23. Classification accuracies of PandaMood feature selection sets, each box plot displays the mean, standard deviation (dotted line) and median (solid line).

4.1.3 Top Five Models

To highlight the GTZAN top feature sets, we selected the top five models in AF.Testing and constructed an aggregate ranking between four relevant properties. Shown in table 18 are the top rankings in artist filtering testing accuracy, testing accuracy, training to testing distance and feature dimensionality. The aggregate rank was computed as the mean rank between each ranking. As a result, we can see that semi-manual selection was ranked first, followed by All features (μ, σ), SB-Entropy(μ, σ), All features (μ) and All features (σ). A top five was not possible since All features (μ) and All features (σ) both shared the fourth aggregate rank position.

TABLE 18. Independent and aggregated ranks between the top 5 GTZAN models.

Aggregated Ranks	Feature Set	Dimensionality Ranks (Value)	AF. Testing Accuracy Ranks (CA)	Testing Accuracy Ranks (CA)	Training - Testing Distance Ranks (CA Distance)	Ranking Average
1st	Top 2 [SB-Entropy(μ, σ) & SB-Flux(μ, σ)]	2nd (40)	3rd (58.81%)	2nd (74.02%)	2nd (11.20%)	2.25
2nd	All Features (μ, σ)	4th (126)	1st (64.57%)	1st (77.83%)	5th (21.17%)	2.75
3rd	SB-Entropy (μ, σ)	1st (20)	5th (53.05%)	5th (66.77%)	1st (6.05%)	3
4th	All Features (μ)	3rd (63)	4th (58.26%)	3rd (73.92%)	3rd (15.33%)	3.25
	All Features (σ)	3rd (63)	2nd (60.25%)	4th (73.39%)	4th (17.57%)	3.25

In table 19 we see the rank aggregation paradigm of table 18 applied to the PandaMood results. We began by selecting the top five models in testing accuracy and aggregated their ranks with dimensionality and training to testing distance rankings. The aggregate rank top model was SB-Entropy(μ) & MFCCs (σ) followed by SB-Entropy (μ, σ), SB-Entropy(μ, σ) & MFCCs (μ, σ), All Features (μ, σ) and All Features (σ). We see that in both tables 18 and 19, semi-manual selection was aggregately ranked first and 'All Features (σ)' last.

TABLE 19. Independent and aggregated ranks between the top 5 PandaMood models.

Aggregated Ranks	Feature Set	Dimensionality Ranks (Value)	Testing Accuracy Ranks (CA)	Training - Testing Distance Ranks (CA Distance)	Ranking Average
1st	Top 2 [SB-Entropy(μ) & MFCCs (σ)]	1st (20)	4th (39.84%)	1st (19.53%)	2.00
2nd	SB-Entropy (μ, σ)	1st (20)	5th (39.51%)	2nd (20.65%)	2.67
	Top 2 [SB-Entropy(μ, σ) & MFCCs (μ, σ)]	3rd (63)	2nd (42.18%)	3rd (25.72%)	2.67
3rd	All Features (μ, σ)	4th (126)	1st (42.41%)	4th (27.79%)	3.00
4th	All Features (σ)	2nd (40)	3rd (40.08%)	5th (27.88%)	3.33

4.1.4 Feature Importance

In this section, we elaborate on the feature importance obtained from automatic and manual feature selection. We construct two tables to showcase the rankings obtained from information gain and semi-manual selection. In each table, we find two rankings one for GTZAN and one for PandaMood. In table 20 we see the rankings obtained from the information gain algorithm within the classification stage. The algorithm was provided with the All Features (μ, σ) set and ranked both statistical summaries and individual feature components. In contrast table 21 shows the individual feature rankings obtained from our classification results. Both IG and the individual feature sets outputted the same dimensionality of selected feature components (20 dimensions), except when the MFCCs (μ, σ) (26 dimensions) were selected in table 21.

TABLE 20. The first 20 feature dimensions (per task) as selected by the information gain feature selection algorithm, (μ) standing for mean values and (σ) for standard deviation values.

Information Gain Feature Rankings	
Music Genre – GTZAN	**Music Mood - PandaMood**
1) SB-Entropy/Octave 6 (μ)	1) SB-Entropy/Octave 4 (μ)
2) SB-Entropy/Octave 5 (μ)	2) SB-Entropy/Octave 5 (μ)
3) SB-Kurtosis/Octave 10 (μ)	3) MFCC 1 (μ)
4) SB-Flux/Octave 9 (σ)	4) SB-Flux/Octave 8 (μ)
5) SB-Skewness/Octave 10 (μ)	5) SB-Kurtosis/Octave 8 (μ)
6) SB-Flux/Octave 9 (μ)	6) SB-Flux/Octave 7 (μ)
7) SB-ZCR/Octave 10 (σ)	7) SB- Skewness /Octave 8 (μ)
8) SB-Flux/Octave 10 (σ)	8) SB-Entropy/Octave 6 (μ)
9) SB-Flux/Octave 8 (μ)	9) SB- Skewness /Octave 4 (μ)
10) SB-Flux/Octave 1 (μ)	10) SB-Entropy/Octave 3 (μ)
11) SB-Entropy/Octave 7 (μ)	11) SB-Entropy/Octave 10 (μ)
12) SB-Entropy/Octave 10 (σ)	12) SB-Kurtosis/Octave 4 (μ)
13) SB-Flux/Octave 8 (σ)	13) SB-Kurtosis/Octave 4 (σ)
14) SB-Flux/Octave 2 (μ)	14) SB- Skewness /Octave 7 (μ)
15) SB-ZCR/Octave 9 (μ)	15) SB- Skewness /Octave 2 (σ)
16) SB- Skewness /Octave 10 (σ)	16) SB-Flux/Octave 7 (σ)
17) SB-Entropy/Octave 4 (μ)	17) SB- Skewness /Octave 5 (μ)
18) SB-Flux/Octave 10 (μ)	18) SB-Kurtosis/Octave 3 (σ)
19) SB-Kurtosis/Octave 9 (μ)	19) SB-Kurtosis/Octave 1 (σ)
20) MFCC 1 (μ)	20) SB-Kurtosis/Octave 5 (μ)

In the GTZAN ranking (table 20) we see that the top feature component was the spectral entropy of the sixth octave (800 – 1600 Hz) summarized with mean values. In perspective, the rankings show 14 (70%) mean summaries, 6 (30%) standard deviation summaries, 19 (95%) sub-band feature components and 1 (5%) MFCC coefficient. The distribution of components and coefficients in descending order was as follows:1) SB-Flux (8 components) 2) SB-Entropy (5 components); 3) SB-Skewness (2 components); 4) SB- ZCR (2 components), 5) SB-Kurtosis (2 components); 6) MFCCs (1 Coefficient). We see that 68% of all sub-band feature components were chosen from SB-Entropy and SB-Flux. In addition, the selected octaves between the components were as follows (Descending order): 1) Octave 10 (7 instances); 2) Octave 9 (4 instances); 3) Octave 8 (2 instances); 4) Octaves 7,6,5,4,2,1 (1 instance each). We see that octave 3 was ignored and that more than 68% of the chosen octaves come from the mid-high end of the spectrum, above the sixth octave (800 – 1600 Hz). Thus, we can consider that the spectral content above 1600 Hz was more relevant to our algorithmic selection. In addition, we can see that the selection algorithm also showed a clear preference for sub-band feature components with mean summary values.

From the PandaMood ranking (table 20), we observe that the top feature was the spectral entropy of the fourth octave (200 - 400 Hz) summarized with mean values. We find that the entire ranking comprised of 15 (75%) mean summaries, 5 (25%) standard deviation summaries, 19 (95%) sub-band components and 1 (5%) MFCC coefficient. Further, we find that the distribution of components and coefficients was as follows (Descending order): 1) SB-Kurtosis (6 components); 2) SB-Entropy (5 components); 3) SB-Skewness (5 components); 4) SB- Flux (3 components); 5) MFCC (1 coefficient). We find that SB-ZCR was not selected and that 16 of 19 (84.21%) sub-band components were selected from SB-Kurtosis, SB-Entropy and SB-Skewness. Furthermore, we find the following component octave distribution (descending order): 1) Octave 4 (4 instances); 2) Octave 5 (3 instances); 3) Octave 7 (3 instances); 4) Octave 8 (3 instances); Octave 3 (2 instances); Octaves 1,2,6,10 (1 instance each). We can discern that octave 9 was ignored and that the four most reoccurring octaves were from the mid-low and mid-high end of the spectrum. Specifically, octaves 4 & 5 (200 – 800 Hz) and 7 & 8 (1600 – 6400 Hz). The findings suggest that algorithmic selection highlighted three frequency bands, mid-low, mid, and high. In addition, we see a strong preference for sub-band feature components and mean value summaries.

Feature Importance Summary

In summary, both task rankings in table 20 highlighted sub-band feature components and mean values. Sub- band entropy components consistently ranked first and second in both task rankings, but sub-band components and octave distributions did not match, except for SB-Entropy/Octave 5 (2nd place in both). The MFCCs were shown to be the least important in GTZAN but achieved third place in PandaMood and second place in table 21 PandaMood. We can see a rough analogy between table 20 and table 21 where SB-Entropy (μ,σ) was found to perform the best in both tasks while the MFCCs performed better in PandaMood. We also see that SB-ZCR ranked worst in table 21 and was completely ignored for Table 20 PandaMood. Both rankings suggest an inconsistent ordering between the two tasks, except for SB-Entropy that occupied the first place in both rankings and tasks.

TABLE 21. Individual feature set rankings of classification accuracy per task.

Individual Feature Classification Accuracy Ranks	
Music Genre – GTZAN	**Music Mood - PandaMood**
1) SB-Entropy (μ, σ)	1) SB-Entropy (μ, σ)
2) SB-Flux (μ, σ)	2) MFCCs (μ, σ)
3) SB-Kurtosis (μ, σ)	3) SB-Skewness (μ, σ)
4) SB-Skewness (μ, σ)	4) SB-Flux (μ, σ)
5) MFCCs (μ, σ)	5) SB-Kurtosis (μ, σ)
6) SB-ZCR (μ, σ)	6) SB-ZCR (μ, σ)

5 DISCUSSIONS

The primary aim of this thesis was to investigate the efficacy of five FDW sub-band features and their statistical summary sub-sets in music genre and music mood classification tasks. To aid our evaluation, we extracted a baseline MFFC feature commonly used in such tasks. In our experimental design, all features were extracted for two music datasets (GTZAN, PandaMood) and were summarized with the mean and standard deviation statistics. The classification stage used three classifiers (SVM, MLR, K-NN) with four feature selection sets. The feature selection sets were as follows; 1) All features 2) Individual features 3) Top two performing individual features (semi-manual selection); 4) Algorithmically selected features (information gain). All feature sets used both the mean and standard deviation statistics. In addition, the 'All features' and the semi-manual sets were further expanded with additional statistical summary combinatorics. In this chapter, we elaborate on the implications of our results and focus on classification performance, classifier overfitting, feature selection, and feature importance. In addition, we highlight the relevant limitations of the study and provide suggestions for future research.

5.1 Classification Performance & Overfitting

In Appendix A we find that the SVM classifier performed the best within the music mood task, whereas MLR performed the best within the music genre task. Antithetically, the K-NN

algorithm underperformed in both tasks. These findings are difficult to explain, what might explain part of the SVM underperformance in music genre might be the lack of optimal SVM hyper-parameters. In contrast, the SVM prominence in the music mood task could be due to the default hyper-parameter values being closer to accuracy improving values in the PandaMood hyperparameter space. All reported accuracies show that the music genre accuracy profile was higher than music mood and that music mood showed a higher tendency to overfit. This accuracy gap is in accordance with the MIREX review in chapter 2. Ultimately, the decreased accuracy profile and the increased overfitting indicators may suggest that the modelling of music mood may be more challenging than music genre.

Between all models, in both tasks, tables 18 and 19 showed that the 'All Features (μ, σ)' set outperformed every other model. Although the testing accuracy was the highest, the model's merit was hindered when we considering overfitting since we found the highest training to testing distance in GTZAN and the second highest in PandaMood. The finding was not surprising given that the set contained the maximum number of features and statistical summaries, part of which might have been redundant. Further support for redundancy came from tables 18 and 19 which showed that semi-manual selection was the first in aggregate rankings and performed similarly to 'All Features (μ, σ)' but with a lower overfitting indicator and a fraction of the dimensionality. Thus, it is plausible to consider that the 'All Features (μ, σ)' set may have had increased overfitting potential mainly due to it's high dimensionality.

Classifier training error scores were absent from the compatible and relevant literature, in which case extensive comparisons with respect to overfitting indicators become problematic. In addition, the use of aggregate rankings helped us to move beyond evaluating models solely on their classification accuracy. Unfortunately, overreliance on classification accuracy is all too common in MIREX, GTZAN and PandaMood literature, effectively limiting the comparative scope of our findings when considering other important aspects such as overfitting.

On the individual feature level, sub-band entropy performed the best as an individual feature and was one of the top five models aggregated in tables 18 and 19. Figure 19 showed that all sub-band features except SB-ZCR outperformed the MFCCs in GTZAN. In contrast, figure

22 showed that only SB-Entropy outperformed the MFCCs in PandaMood. In both tasks, the MFCCs demonstrated the highest tendency to overfit the data. Our performance ranks followed and expanded previous work (M. A. Hartmann, 2011) where the SB-Flux outperformed the MFCCs in GTZAN. Our findings suggest that SB-Entropy is suitable for both tasks, whereas the MFCCs could have a better supporting role in music mood classification.

Between feature selection approaches (figures 20 and 23) we found that the semi-manual feature selection (top 2 feature sets) was able to outperform algorithmic feature selection (information gain). These findings were especially surprising when we consider the common notion and the work advocating the efficacy of automatic feature selection algorithms (Guyon & Elisseeff, 2003; Weston et al., 2001). Our results may be partly explained from our semi-manual ranking approach. We can conceptualize the semi-manual selection method as wrapper method (Guyon & Elisseeff, 2003) with a fixed feature set specification since we considered only individual features with both the mean and standard deviation. In contrast, information gain does not consider sets but every feature dimension. Besides, the semi-manual selection employed the classification stage to rank the merit of feature sets, whereas information gain by design, does not. Thus, it can be plausible to consider that the classifier-based rankings and the fixed set approach may have played a role in the difference that was observed.

Exclusively for GTZAN, we found that the GTZAN literature does not employ fault filtering and artist filtering, except in some works (Jeong & Lee, 2016; Kereliuk, Sturm, & Larsen, 2015; J. Lee, Park, Kim, & Nam, 2018; Medhat, Chesmore, & Robinson, 2017; Park, Lee, Park, et al., 2017; Pons & Serra, 2018; Sturm, 2013b, 2014b). Given that previous studies showed that GTZAN faults (Sturm, 2014b) and the lack of artist filtering (Jeong & Lee, 2016; Kereliuk et al., 2015; Medhat et al., 2017; Sturm, 2014b) can inflate classification accuracy, this brings considerable doubts about the validity and comparability of non-fault, and non-artist filtered models.

Figures 18, 19 and 20 showed that all artist filtered models performed considerably lower than the non-filtered models. These findings are consistent with previous findings (Jeong & Lee, 2016; Kereliuk et al., 2015; Medhat et al., 2017; Sturm, 2014b). In addition, we find that

artist filtered models did not rank analogously to their non-filtered models, also found in previous works (Jeong & Lee, 2016; Medhat et al., 2017; Sturm, 2014b) strongly suggesting a somewhat inconsistent classifier behavior. We find that the artist filtered scores in table 18 were in line with a portion of previous findings (Jeong & Lee, 2016; Kereliuk et al., 2015; Medhat et al., 2017; Pons & Serra, 2018; Sturm, 2013b, 2014) and considerably lower than the remainder (J. Lee & Nam, 2017b; J. Lee et al., 2018; Park, Lee, Park, et al., 2017). Direct comparisons are problematic because we employ only classification accuracies, an extended artist filtering (deleted other than first versions), and our artist to fold distribution was automatically generated (with Scikit-learn). Despite any comparative limitations, we find that all works do not report any overfitting indicators, as such, any comparisons concerning overfitting cannot be made.

Exclusively for PandaMood, we found a limited amount of publications (Baniya, Hong, & Lee, 2015; Panda, Malheiro, & Paiva, 2018; Panda et al., 2013; Ren, Wu, & Jang, 2015) of which two (Baniya et al., 2015; Ren et al., 2015) allow for comparing with our results. The main reason for the incompatibility was that the remaining works (Panda et al., 2018, 2013) did not report classification accuracy, whilst all works did not report training errors. Given these inconsistencies in the experimental setup, it becomes difficult to ascertain a transparent state of evaluation for this dataset, especially in overfitting terms. Nevertheless, we find that 'SB-Entropy(μ, σ) & MFCCs (μ, σ)' performed similarly to most models found in Ren et al., (2015) (except those exceeding 400 dimensions), outperforming most models in average accuracy, but not in standard deviation. This was not the case in comparison to the model found in Baniya et al., (2015). Despite this comparative scope, it is problematic to consider the competitiveness of all works between themselves and to our work, that is because no other work reported overfitting indicators.

5.2 Feature Importance

In table 20 we observed that 95% of all automatically selected feature components belonged to sub-band features mostly summarized with the mean statistic. This finding was not surprising considering the substantial amount of sub-band features in the feature pool. For GTZAN our findings differ from previous findings (M. A. Hartmann, 2011), where standard deviation values have been deemed the most important. The reason behind this difference

may lie in the use of different feature selection methods and our lack of feature selection aggregate rankings.

Concerning individual feature importance, we find that tables 18 and 19 show that SB-Entropy ranked first in both tasks and across both feature-selection approaches (semi-manual and automatic). In automatic selection (table 20) we see that SB-Entropy in the 6th octave was most relevant for music genre as opposed to SB-Entropy in the 4th octave for music mood. In addition, we find that SB-Flux and SB-Entropy were the most frequently selected feature components for GTZAN, versus SB-Kurtosis and SB-Entropy for PandaMood. These findings further support the efficacy of SB-Entropy in both classification tasks.

In table 20 we find that the frequency bands of the selected sub-band feature components varied between the tasks. In GTZAN the majority of sub-band components were selected between the 6th (800 – 1600 Hz) and 10th octave (12800 – 22050 Hz), where the 10th octave had the maximum recurrence rate. This finding shows a particular focus on the high and mid-high end of the spectrum, which is challenging to explain. This finding differs from previous findings (M. A. Hartmann, 2011) were all octaves were relevant. In contrast, the PandaMood rankings focused in octaves 4 - 5 (200 – 800 Hz) and octaves 7 - 8 (1600 – 6400 Hz), where the 4th octave had the highest recurrence rate. The PandaMood octave focus is contradictory to GTZAN and suggests that different spectral regions might be more relevant to the tasks.

5.3 Chance Levels

A fundamental consideration for evaluating classification performance are chance levels, and the extend of model performance divergence from the chance level baseline. For classes with unevenly distributed finite data the default dummy-classifier in the Scikit-learn library (Pedregosa et al., 2011) allows to estimate baseline chance levels while respecting class data distribution. We obtain, 12.85% for GTZAN (10 classes) and 20.71% for PandaMood (5 classes). These values are close to what is obtained when assuming uniform class distribution and infinite data (10% for 10 classes, 20% for 5 classes). We find that GTZAN models performed higher than PandaMood models with respect to the chance level base line, despite having a lower baseline.

5.4 Limitations

In this section, we discuss the various technical and methodological limitations that can affect the outcome of the study. These limitations pertain to the overall evaluation approach, the music datasets, the feature selection rankings, the aggregate top model rankings, the feature extraction parameters and classifier overfitting.

5.4.1 Statistical Summaries (Bag-of-Frames)

Training a classifier model with feature vector statistical summaries is often called a 'bag of frames' approach. This approach is a big limitation in that it can produce identical models from identical randomly scrambled individual audio segments (Aucouturier, 2008). That is not surprising since any identical temporal sequence randomly rearranged will produce the same summary statistics as the original sequence. In certain problems, where temporal dynamics are irrelevant, this approach poses no limitation, but in the case of music, it raises questions about the 'musicality' of the trained models. One arbitrary analogy of the 'un-musicality' of such models, is that a person listening to randomly rearranged music segments may classify them as 'experimental', whereas a model unaware of temporal dynamics would not. Therefore, it would be a positive direction for future research to intergrade temporal dynamics in the training process.

5.4.2 No Validation Set

We employed no validation set, given the relatively small size of our two datasets (in contrast to MIREX), a percentage of data withheld from training and testing would further raise questions of learning efficacy. In the case where more data would be available, a validation set would improve evaluation, especially for accessing testing set overfitting. Therefore, focusing on expanding the current datasets would help to facilitate a cross-validation design with validation sets.

5.4.3 No Artist Filtering for PandaMood

The effects of artist filtering were shown for music genre but not for the music mood tasks since no such filter was available. This limitation restricts the scope of analysis for

PandaMood since we have seen that artist filtered based systems can behave differently than non-filtered ones. Investigating such effects for music mood may be relevant.

5.4.4 No Cross-Dataset Validation

As described by Bogdanov, Porter, Herrera, and Serra (2016), the cross-evaluation of models built on different datasets can help in accessing generalization capabilities. No such validation methods were used in the current study. In an ideal paradigm, compatible data from different datasets could be used to cross-validate models trained with each dataset. Therefore, it is expected that this approach would increase the available data for evaluation and help to detect overfitting.

5.4.5 GTZAN Artist Filter

We find four limitations in the artist filter use of GTZAN: 1) Unfiltered models are difficult to compare to filtered ones. This is not surprising considering the differences in the allocation of the data between training and testing sets; 2) Only a limited amount of folds is possible in filtered models. Many GTZAN classes are overpopulated with small amounts of artists (Sturm, 2014b), because of this, creating more than two folds could undermine evaluation validity; 3) After applying artist filtering, data to class allocation becomes disproportionate. This leads to an un-stratified and uneven learning process that may need some type of normalization; 4) Only a limited amount of AF-GTZAN studies have been made. This is especially limiting when it comes to methodological and result comparisons.

5.4.6 GTZAN Fault Filtering Limitations

We found three critical limitations when fault filtering the GTZAN data: 1) Fault filtering unbalanced the data distribution between classes. This occurred because nearly 10% (97 files) from the original data were deleted; 2) Comparisons with non-fault filtered models become difficult, because GTZAN faults (replicas, distortions, etc.) have a performance inflationary effect (Sturm, 2014b). Thus, it becomes difficult to ascertain how non-filtered models perform given that their accuracy profiles could be different had the faults been deleted; 3) Comparisons between fault filtered studies becomes difficult if the fault filtering specifications are different.

5.4.7 Audio Window Decomposition

In our study, we introduced and employed the FDW method as opposed to conventional single size windowing. It is, therefore, an open question as to how the two windowing methods compare with each other and between different problem domains. Future work may focus on using FDW in other evaluation tasks and in comparison to conventional windowing.

5.4.8 Confusion Quality Analysis

In tables 18 and 19 we constructed aggregate ranks from multiple rankings of interest. Individual ranks help to differentiate between models, but they do not include classifier error/confusion quality. Low error quality refers to extreme classifier confusions as opposed to human expert confusions, high error quality is maximal when confusions approximate expert ‘intuitive’ confusions. Although error/confusion quality is problematic to formalize and implement such an approach could help to select models that do not generate extreme mistakes. Ultimately, the tolerance for the quality of errors will depend on the use case.

5.4.9 Feature Combinatorics

In our study, we focused on one part out of all the possible combinatorics within and between feature sets and statistical summaries. It is unknown if other combinations could match, underperform or outperform our current selections.

5.4.10 No Content Based FDW

FDW operates irrespective of sub-band signal content, which means that window sizes are computed only concerning sub-band central frequencies. After a filterbank is specified the resulting FDW window specifications will not change. In it, of itself, this may not be considered a direct limitation, but it may be plausible to hypothesize a further potential benefit to classification from content level adaptive windowing for each filtered signal within each sub-band. Such an approach would necessarily produce multiple window size specifications for different sub-band signal contents. Given that statistical summaries can be extracted, the variance between window sizes would not be an issue for performing classification. One such approach may be the calculation of an average spectral centroid for

each sub-band signal (instead of central frequency), in turn the centroid value may be feed into FDW to produce the window size.

5.4.11 Aggregate Rankings

In our results section we constructed two aggregate top 5 model rankings, the aggregation process consisted of the average rank among many individual rankings of interest (testing score, dimensionality, etc.). The main limitation of this approach was that each ranking was considered of equal importance since the weights for each ranking could not be collected. Specifying fixed weights for individual rankings is problematic since the relative importance of each ranking of interest can heavily depend on the problem and the intended use case. Moreover, even in the case where particular rankings are considered, it can be difficult to formalize and validate such weighted rankings.

5.4.12 Spectral Features & Music Mood Classification

In our experiments, we employed spectral features in music mood classification, yet as seen in MIREX, music mood is often modelled with diverse feature groups, and not entirely with spectral features. Future research may focus in incorporating and evaluating our feature set with non-spectral features.

5.4.13 Overfitting Indicators

Overfitting is a critical consideration for any machine learning system, regardless of the problem domain. The importance of detecting and dealing with overfitting cannot be stressed enough. In our study, various models showed high overfitting potential, especially for the PandaMood dataset. Although extensive training to testing set divergence indicates a tendency to overfit, it is unclear how large should this distance be, before considering the phenomena is severe. Also, it is empirically unclear if the critical divergence range shifts between different problem domains or sub-domains. The underlying phenomenon is more accessible to detect when a validation set is used, especially concerning the testing set.

5.4.14 SVM Hyper Parameter Optimization

In our study, SVM grid-search hyperparameter optimization did not converge to acceptable values and thus we focused on using default values. Given enough time, a larger grid or a different approach altogether (Bayesian optimization [Bergstra, Komer, Eliasmith, Yamins, & Cox, 2015], random search [James & Bengio, 2012], etc.) the optimization process might have resulted in acceptable values.

5.5 Conclusions

In conclusion, we had seven primary outcomes in our study; Firstly, all accuracy scores indicated a higher performance profile for the music genre task than music mood. Secondly, sub-band entropy ranked first in both feature selection methods (semi-manual and automatic), outperformed the MFCCs and all SB-Features in both music genre and music mood tasks. In GTZAN each sub-band feature (except SB-ZCR) outperformed the MFCCs. Third, all sub-band features displayed a smaller tendency to overfit than the MFCCs. Fourth, semi-manual selection (Top 2 features) outperformed automatic feature selection (information gain) in both tasks. Fifth, semi-manual selection outranked all other models when considering an average rank between testing accuracy, artist filter testing accuracy, dimensionality and overfitting indicators. Fifth, music genre artist filtered models performed lower than non-filtered models with feature set rankings differing between the two. Six, music genre automatic artist filter partitioning performed similarly to studies with manual partitioning. Seven, information gain feature selection focused on different spectral regions for each task, octave 10 (12800 – 22050 Hz) was most relevant for music genre and octave 4 (200 – 400 Hz) for music mood.

Future research could focus on the efficacy of the sub-band features, and especially the sub-band entropy in other classification tasks where spectral features are relevant; Such tasks include audio tag classification, artist identification, audio and music similarity estimation, structural segmentation, audio fingerprinting, and speech analysis. In addition, SB-Entropy, SB-Skewness, SB-Kurtosis and SB-ZCR have not been perceptually validated which remains an open question. An additional approach for future research would be the development of the sub-band concept with different features, filter designs, windowing methods and filter orders. Such an approach might bring forth potentially novel and beneficial features. Finally, other

cultural contexts such as classification tasks with non-western music could further broaden the scope of evaluation and provide an extended perspective for the sub-band features.

Future research on the dataset level (GTZAN, PandaMood) would greatly benefit from a standardized approach to evaluation. The state of evaluation is deeply inconsistent within and between each task. The central gap in evaluation derives from the lack of, fault checking, common figures of merit, reported training errors, aggregate ranks and artist filtering. These discrepancies halt steady development in both tasks and limit the comparative scope between classification systems.

References

Alluri, V. (2012). *Acoustic, neural, and perceptual correlates of polyphonic timbre*. University of Jyväskylä. Retrieved from https://jyx.jyu.fi/dspace/bitstream/handle/123456789/38121/9789513946548.pdf?sequence=1

Alluri, V. (2015). *Timbre [Oral Presentation]*. Jyväskylä, Finland.

Alluri, V., & Toiviainen, P. (2009). In search of perceptual and acoustical correlates of polyphonic timbre. In *Conference of European Society for the Cognitive Sciences of Music*. University of Jyväskylä.

Alluri, V., & Toiviainen, P. (2010). Exploring Perceptual and Acoustical Correlates of Polyphonic Timbre. *Music Perception*, *27*(3), 223–242. https://doi.org/10.1525/mp.2010.27.3.223

Alluri, V., & Toiviainen, P. (2012). Effect of Enculturation on the Semantic and Acoustic Correlates of Polyphonic Timbre. *Music Perception: An Interdisciplinary Journal*, *29*(3), 297–310. https://doi.org/10.1525/mp.2012.29.3.297

Alluri, V., Toiviainen, P., Jääskeläinen, I. P., Glerean, E., Sams, M., & Brattico, E. (2012). Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm. *NeuroImage*, *59*(4), 3677–3689. https://doi.org/10.1016/j.neuroimage.2011.11.019

Aucouturier, J. (2008). *Splicing : A Fair Comparison Between Machine and Human on a Music Similarity Task*. Citeseer. Retrieved from https://pdfs.semanticscholar.org/f64f/4753926d534c99e37bbe01ecffc67405a95b.pdf

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 627–635.

Baniya, B. K., Hong, C. S., & Lee, J. (2015). Nearest multi-prototype based music mood classification. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science, ICIS 2015 - Proceedings* (pp. 303–306). IEEE. https://doi.org/10.1109/ICIS.2015.7166610

Barbedo, J. G. A., & Lopes, A. (2007). Automatic genre classification of musical signals. *Eurasip Journal on Advances in Signal Processing*, *2007*(1), 157. https://doi.org/10.1155/2007/64960

Belin, P., Fillion-Bilodeau, S., & Gosselin, F. (2008). The Montreal Affective Voices: a validated set of nonverbal affect bursts for research on auditory affective processing. *Behavior Research Methods*, *40*(2), 531–539.

Bergstra, J., Casagrande, N., & Eck, D. (2005). *Two algorithms for timbre and rhythm based multiresolution audio classification*. Retrieved from https://www.music-ir.org/mirex/abstracts/2005/bergstra.pdf

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: A Python library for model selection and hyperparameter optimization. *Computational Science and Discovery*, *8*(1). https://doi.org/10.1088/1749-4699/8/1/014008

Bigand, E., Vieillard, S., Madurell, F., Marozeau, J., & Dacquet, A. (2005). Multidimensional scaling of emotional responses to music: The effect of musical expertise and of the duration of the excerpts. *Cognition and Emotion*, *19*(8), 1113–1139. https://doi.org/10.1080/02699930500204250

Bogdanov, D., Porter, A., Herrera, P., & Serra, X. (2016). Cross-collection evaluation for music classification tasks. In *Proceedings of the International Society for Music Information Retrieval Conference* (Vol. 16, pp. 379–385). Universitat Pompeu Fabra. Retrieved from https://repositori.upf.edu/bitstream/handle/10230/33061/Bogdanov_ISMIR2016_cros.pdf?sequence=1&isAllowed=y

Böhning, D. (1992). Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, *44*(1), 197–200.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144–152). Assn for Computing Machinery. https://doi.org/10.1145/130385.130401

Bracewell, R. N., & Bracewell, R. N. (1986). *The Fourier transform and its applications* (Vol. 31999). McGraw-Hill New York.

Bradley, M. M., & Lang, P. J. (2000). Affective reactions to acoustic stimuli. *Psychophysiology*, *37*(2), 204–215. https://doi.org/10.1017/S0048577200990012

Burger, B. (2013). *Move the way you feel: Effects of musical features, perceived emotions, and personality on music-induced movement*. University of Jyväskylä. Retrieved from https://jyx.jyu.fi/dspace/bitstream/handle/123456789/42506/978-951-39-5466-6_vaitos07122013.pdf?sequence=1

Caclin, A., McAdams, S., Smith, B. K., & Winsberg, S. (2005). Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones. *The Journal of the Acoustical Society of America*, *118*(1), 471–482. https://doi.org/10.1121/1.1929229

Cao, C., & Li, M. (2008). *Thinkit Audio Genre Classification System*. Citeseer. Retrieved from https://www.music-ir.org/mirex/abstracts/2008/mirex08_genre_CC.pdf

Cao, C., & Li, M. (2009). Thinkit'S Submissions for Mirex2009 Audio Music Classification and Similarity Tasks. In *Proceedings of the International Symposium on Music Information Retrieval* (pp. 1–3). Citeseer. https://doi.org/10.1080/00048623.2013.799034

Celma, O. (2010). Music recommendation. In *Music recommendation and discovery* (pp. 43–85). Springer.

Celma, Ò., Herrera, P., & Serra, X. (2006). Bridging the Music Semantic Gap. In *Workshop on Mastering the Gap: From Information Extraction to Semantic Representation* (Vol. 187). Budva, Montenegro: CEUR. Retrieved from http://hdl.handle.net/10230/34294

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Transfer learning for music classification and regression tasks. *ArXiv Preprint ArXiv:1703.09179*. Retrieved from http://arxiv.org/abs/1703.09179

Collier, G. L. (2007). Beyond valence and activity in the emotional connotations of music. *Psychology of Music*, *35*(1), 110–131. https://doi.org/10.1177/0305735607068890

Cunningham, S. J., Bainbridge, D., & Falconer, A. (2006). "More of an art than a science": Supporting the creation of playlists and mixes. In *ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October 2006, Proceedings* (pp. 240–245). University of Victoria. Retrieved from http://researchcommons.waikato.ac.nz/handle/10289/77

Cunningham, S. J., Jones, M., & Jones, S. (2004). Organizing digital music for use: an examination of personal music collections. In *ISMIR 2004, 5th International Conference on Music Information Retrieval, Barcelona, Spain, October 10-14, 2004, Proceedings*. Pompeu Fabra University. https://doi.org/10.1073/pnas.0914115107

Daubechies, I. (1990). The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, *36*(5), 961–1005. https://doi.org/10.1109/18.57199

Demšsar, J., Curk, T., Erjavec, A., Hočevar, T., Milutinovič, M., Možzina, M., … Zupan, B. (2013). Orange: data mining toolbox in Python. *The Journal of Machine Learning Research*, *14*(1), 2349–2353. https://doi.org/10.1088/0957-4484/23/3/035606

Dowling, W. J., & Harwood, D. L. (1986). *Music cognition*. Academic Press. Retrieved from https://www.sciencedirect.com/book/9780080570686/music-cognition

Dudani, S. A. (1976). The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-6*(4), 325–327. https://doi.org/10.1109/TSMC.1976.5408784

Eerola, T., Ferrer, R., & Alluri, V. (2012). Timbre and Affect Dimensions: Evidence from Affect and Similarity Ratings and Acoustic Correlates of Isolated Instrument Sounds. *Music Perception: An Interdisciplinary Journal*, *30*(1), 49–70. https://doi.org/10.1525/mp.2012.30.1.49

Eerola, T., & Vuoskoski, J. K. (2011). A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, *39*(1), 18–49. https://doi.org/10.1177/0305735610362821

Eerola, T., & Vuoskoski, J. K. (2013). A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli. *Music Perception: An Interdisciplinary Journal*, *30*(3), 307–340. https://doi.org/10.1525/mp.2012.30.3.307

Egenhofer, M. J., Giudice, N., Moratz, R., & Worboys, M. (2011). *Spatial information theory : 10th International Conference, COSIT 2011, Belfast, ME, USA, September 12-16, 2011, proceedings* (Vol. 6899). Springer. Retrieved from https://books.google.fi/books?hl=en&lr=&id=uD3HbuMeo28C&oi=fnd&pg=PP2&dq=Spatial+Information+Theory:+10th+International+Conference,+COSIT+2011,+Belfast,+ME,+USA&ots=PDOisBhtnM&sig=50wrk-m9TxYUYOKjjMs-Ac21YaA&redir_esc=y#v=onepage&q=Spatial Information The

Ekman, P. (1971). Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation*. University of Nebraska Press.

Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, *6*(3–4), 169–200.

Ekman, P., & Cordaro, D. (2011). What is meant by calling emotions basic. *Emotion Review*, *3*(4), 364–370. https://doi.org/10.1177/1754073911410740

Ekman, P. E., & Davidson, R. J. (1994). *The nature of emotion: Fundamental questions.* Oxford University Press.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second International Conference on Knowledge Discovery & Data Mining: Proceedings* (Vol. 96, pp. 226–231). https://doi.org/10.1.1.71.1980

Flexer, A., & Schnitzer, D. (2009). Album and artist effects for audio similarity at the scale of the web. In *Proceedings of the 6th Sound and Music Computing Conference* (Vol. 15, pp. 59–64). Porto - Portugal: Citeseer.

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

Frijda, N. H. (2007). What might emotions be? Comments on the Comments. *Social Science Information*, *46*(3), 433–443. https://doi.org/10.1177/0539018407046003011 2

genre, n.: Oxford English Dictionary. (2016). Retrieved May 27, 2018, from http://www.oed.com/view/Entry/77629?redirectedFrom=genre#eid

genre | Origin and meaning of genre by Online Etymology Dictionary. (2017). Retrieved May 27, 2018, from https://www.etymonline.com/word/genre

Gouyon, F., Pachet, F., & Delerue, O. (2000). On the use of Zero-Crossing rate for an application of classification of percussive sounds. In *Proceedings of the International Conference on Digital Audio Effects* (pp. 3–8).

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*(3), 1157–1182. https://doi.org/10.1016/j.aca.2011.07.027

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1995). *Neural Network Design*. *Boston Massachusetts PWS* (Vol. 2). Pws Pub. Boston.

Hamel, P. (2011). *Pooled Features Classification Mirex 2011 Submission*. Retrieved from https://www.music-ir.org/mirex/abstracts/2011/PH2.pdf

Handel, S. (1995). Timbre Perception and Auditory Object Identification. *Hearing*, *2*, 425–461. https://doi.org/10.1016/B978-012505626-7/50014-5

Hartmann, M. A. (2011). *Testing a spectral-based feature set for audio genre classification*. University of Jyväskylä. Retrieved from jyx.jyu.fi/handle/123456789/36531

Hartmann, M., Saari, P., Toiviainen, P., & Lartillot, O. (2013). Comparing timbre-based features for musical genre classification. In *Proceedings of the International Conference on Sound and Music Computing* (pp. 707–714). Retrieved from http://smcnetwork.org/system/files/Comparing%2520Timbre-based%2520Features%2520for%2520Musical%2520Genre%2520Classification.pdf

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support Vector Machines. *Feature Selection and Ensemble Methods for Bioinformatics*, *13*(4), 68–116. https://doi.org/10.4018/978-1-60960-557-5.ch007

Hevner, K. (1936). Experimental studies of the elements of expression in music. *American Journal of Psychology*, *48*(2), 246–268. https://doi.org/10.2307/1415746

Hoefle, S., Engel, A., Basilio, R., Alluri, V., Toiviainen, P., Cagy, M., & Moll, J. (2018). Identifying musical pieces from fMRI data using encoding and decoding models. *Scientific Reports*, *8*(1), 2266. https://doi.org/10.1038/s41598-018-20732-3

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.

Hu, X., & Downie, J. S. (2007). Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata. In *8th International Society for Music Information Retrieval Conference – ISMIR 2007* (pp. 67–72). https://doi.org/10.1.1.205.8782

Ilie, G., & Thompson, W. F. (2006). A Comparison of Acoustic Cues in Music and Speech for Three Dimensions of Affect. *Music Perception*, *23*(4), 319–330. https://doi.org/10.1525/mp.2006.23.4.319

Izard, C. E. (2007). Basic Emotions, Natural Kinds, Emotion Schemas, and a New Paradigm. *Perspectives on Psychological Science*, *2*(3), 260–280. https://doi.org/10.1111/j.1745-6916.2007.00044.x

Izard, C. E., Ackerman, B. P., Schoff, K. M., & Fine, S. E. (2000). Self-Organization of Discrete Emotions, Emotion Patterns, and Emotion-Cognition Relations. In M. D. Lewis & I. Granic (Eds.), *Emotion, Development, and Self-Organization* (pp. 15–36). Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511527883.003

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/J.PATREC.2009.09.011

James, B., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305. Retrieved from http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

Jeong, I.-Y., & Lee, K. (2016). Learning temporal features using a deep neural network and its application to music genre classification. In *Proceedings of the 17th International Society for Music Information Retrieval Conference* (pp. 434–440). Retrieved from https://www.researchgate.net/profile/Il_Young_Jeong/publication/305683876_Learning_temporal_features_using_a_deep_neural_network_and_its_application_to_music_genre_classification/links/5799a27c08aec89db7bb9f92.pdf

Juslin, P. N., Karlsson, J., Lindström, E., Friberg, A., & Schoonderwaldt, E. (2006). Play it again with feeling: Computer feedback in musical communication of emotions. *Journal of Experimental Psychology: Applied*, *12*(2), 79–95. https://doi.org/10.1037/1076-898X.12.2.79

Juslin, P. N., & Laukka, P. (2003). Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological Bulletin*, *129*(5), 770–814. https://doi.org/http://dx.doi.org/10.1037/0033-2909.129.5.770

Juslin, P. N., & Laukka, P. (2004). Expression, Perception, and Induction of Musical Emotions: A Review and a Questionnaire Study of Everyday Listening. *Journal of New Music Research*, *33*(3), 217–238. https://doi.org/10.1080/0929821042000317813

Juslin, P. N., & Vastfjall, D. (2008). Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and Brain Science*, *31*(5), 559–575. https://doi.org/10.1017/S0140525X08005293

Kereliuk, C., Sturm, B. L., & Larsen, J. (2015). Deep learning and music adversaries. *IEEE Transactions on Multimedia*, *17*(11), 2059–2071.

Knees, P., & Schedl, M. (2013). Music similarity and retrieval. In *Proceedings of the 36th international conference on Research and development in information retrieval* (p. 1125). Retrieved from http://www.cp.jku.at/tutorials/sigir2013.html

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence* (Vol. 2, pp. 1137–1143). https://doi.org/10.1067/mod.2000.109031

Krishnapuram, B., Carin, L., Figueiredo, M. A. T., & Hartemink, A. J. (1992). Sparse Multinomial Logistic Regression: Fast Algorithm and Generalization Bounds. *IEEE Transactions on Pattern Analysis and Machine Learning*, *27*(6), 957–968.

Lartillot, O., Eerola, T., Toiviainen, P., & Fornari, J. (2008). MULTI-FEATURE MODELING OF PULSE CLARITY: DESIGN, VALIDATION AND OPTIMIZATION. *ISMIR*.

Lartillot, O., Toiviainen, P., & Eerola, T. (2008). A Matlab Toolbox for Music Information Retrieval. In *International Conference on Digital Audio Effects* (pp. 261–268). Bordeaux, FR. https://doi.org/10.1007/978-3-540-78246-9_31

Laukka, P., Juslin, P., & Bresin, R. (2005). A dimensional approach to vocal expression of emotion. *Cognition & Emotion*, *19*(5), 633–653. Retrieved from http://www.tandfonline.com/doi/citedby/10.1080/02699930441000445

Le, Q. V. (2015). A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks. *Tutorial*. Retrieved from http://robotics.stanford.edu/~quocle/tutorial2.pdf

Lee, J. H., Choi, K., Hu, X., & Downie, J. H. (2013). K-Pop genres: A cross-cultural exploration. In *Proceedings of the 14th Conference of the International Society for Music Information Retrieval*.

Lee, J., & Nam, J. (2017a). Multi-Level and Multi-Scale Feature Aggregation Using Pretrained Convolutional Neural Networks for Music Auto-Tagging. *IEEE Signal Processing Letters*, *24*(8), 1208–1212. https://doi.org/10.1109/LSP.2017.2713830

Lee, J., & Nam, J. (2017b). Multi-level and multi-scale feature aggregation using sample-level deep convolutional neural networks for music classification. *ArXiv Preprint ArXiv:1706.06810*.

Lee, J., Park, J., Kim, K. L., & Nam, J. (2017). Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms. *ArXiv Preprint ArXiv:1703.01789*. https://doi.org/10.1007/s10694-012-0265-x

Lee, J., Park, J., Kim, K., & Nam, J. (2018). SampleCNN: End-to-End Deep Convolutional Neural Networks Using Very Small Filters for Music Classification. *Applied Sciences*, *8*(2), 150. https://doi.org/10.3390/app8010150

Lee, J., Park, J., Nam, J., Kim, C., Kim, A., Park, J., & Ha, J.-W. (2017). *Cross-Cultural Transfer Learning Using Sample-Level Deep Convolutional Neural Netowrks*. *Music Information Retrieval Evaluation eXchange (MIREX)*.

Leman, M., Vermeulen, V., De Voogdt, L., Moelants, D., & Lesaffre, M. (2005). Prediction of musical affect using a combination of acoustic structural cues. *Journal of New Music Research*, *34*(1), 39–67. https://doi.org/10.1080/09298210500123978

Lidy, T., & Schindler, A. (2016). *Parallel Convolutional Neural Networks for Music Genre and Mood Classification*. Retrieved from https://www.music-ir.org/mirex//abstracts/2016/LS1.pdf

Lie, J. (2012). What is the K in K-pop? South Korean popular music, the culture industry, and National Identity. *Korea Observer*, *43*(3), 339–363. https://doi.org/10.1016/j.conbuildmat.2015.07.047

Liu, H., & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining* (Vol. 454). Springer Science & Business Media. https://doi.org/10.1007/978-1-4615-5689-3

Logan, B. (2000). Mel Frequency Cepstral Coefficients for Music Modeling. In *International Symposium on Music Information Retrieval* (Vol. 28, p. 5). https://doi.org/10.1.1.11.9216

Luxburg, U. von, & Schölkopf, B. (2011). Statistical Learning Theory: Models, Concepts, and Results. In *Handbook of the History of Logic* (Vol. 10, pp. 651–706). Elsevier. https://doi.org/10.1016/B978-0-444-52936-7.50016-1

Mandel, M. I., & Ellis, D. (2006). Song-level features and SVM for music classification. In *International Symposium on Music Information Retrieval*.

Marozeau, J., & de Cheveigné, A. (2007). The effect of fundamental frequency on the brightness dimension of timbre. *The Journal of the Acoustical Society of America*, *121*(1), 383–387. https://doi.org/10.1121/1.2384910

Martens, D., Baesens, B., & Gestel, T. Van. (2009). Decompositional rule extraction from support vector machines by active learning. *IEEE Transactions on Knowledge and Data Engineering*, *21*(2), 178–191. https://doi.org/10.1109/TKDE.2008.131

Matityaho, B., & Furst, M. (1995). Neural network based model for classification of music type. In *Proc. Eighteenth Convention of Electrical and Electronics Engineers in Israel* (p. 4.3.4/1--4.3.4/5). https://doi.org/10.1016/j.ejrad.2013.09.030

McAdams, S. (1993). Recognition of auditory sound sources and events. *Thinking in Sound: The Cognitive Psychology of Human Audition*, 146–198. https://doi.org/10.1093/acprof

Mcadams, S., & Giordano, B. L. (2014). The perception of musical timbre. *The Oxford Handbook of Music Psychology*, (February), 72–80. https://doi.org/10.1093/oxfordhb/9780199298457.013.0007

McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., & Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, *58*(3), 177–192. https://doi.org/10.1007/BF00419633

McFee, B., Bertin-Mahieux, T., Ellis, D. P. W., & Lanckriet, G. R. G. (2012). The million song dataset challenge. In *Proceedings of the 21st international conference companion on World Wide Web* (Vol. 2, p. 909). https://doi.org/10.1145/2187980.2188222

Medhat, F., Chesmore, D., & Robinson, J. (2017). Masked conditional neural networks for audio classification. In *International Conference on Artificial Neural Networks* (pp. 349–358).

Mercer, J. (1909). Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *209*(441–458), 415–446. https://doi.org/10.1098/rsta.1909.0016

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, *116*, 374–388. Retrieved from http://www.haskins.yale.edu/sr/SR047/SR047_07.pdf

Misra, H., Ikbal, S., Bourlard, H., & Hermansky, H. (2004). Spectral entropy based feature for robust ASR. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on* (Vol. 1, p. I--193).

Mulligan, K., & Scherer, K. R. (2012). Toward a working definition of emotion. *Emotion Review*, *4*(4), 345–357.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1717–1724).

Pachet, F., & Aucouturier, J.-J. (2004). Improving timbre similarity: How high is the sky. *Journal of Negative Results in Speech and Audio Sciences*, *1*(1), 1–13. Retrieved from http://www.csl.sony.fr/downloads/papers/uploads/aucouturier-04b.pdf

Paiva, R. P. (2012). *MIREX 2012 : MOOD CLASSIFICATION TASKS SUBMISSION*. Retrieved from https://www.music-ir.org/mirex/abstracts/2012/PP5.pdf

Panda, R., Malheiro, R. M., & Paiva, R. P. (2018). Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*. https://doi.org/10.1109/TAFFC.2018.2820691

Panda, R., Malheiro, R., Rocha, B., Oliveira, A. P., & Paiva, R. P. (2013). Multi-Modal Music Emotion Recognition: A New Dataset, Methodology and Comparative Analysis. In *10th International Symposium on Computer Music Multidisciplinary Research* (pp. 570–582). Retrieved from https://eden.dei.uc.pt/~ruipedro/publications/Conferences/CMMR2013_MultiModal.pdf

Panda, R., Rui, B. R., & Paiva, P. (2014). *MIREX 2014: MOOD CLASSIFICATION TASKS SUBMISSION*. Retrieved from https://www.music-ir.org/mirex/abstracts/2014/PP1.pdf

Park, J., Lee, J., Nam, J., Park, J., & Ha, J.-W. (2017). *REPRESENTATION LEARNING USING ARTIST LABELS FOR AUDIO CLASSIFICATION TASKS*. Retrieved from https://www.music-ir.org/mirex//abstracts/2017/PLNPH1.pdf

Park, J., Lee, J., Park, J., Ha, J.-W., & Nam, J. (2017). Representation Learning of Music Using Artist Labels. *ArXiv Preprint ArXiv:1710.06648*. Retrieved from http://arxiv.org/abs/1710.06648

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.1007/s13398-014-0173-7.2

Peeters, G. (2008). A generic training and classification system for MIREX08 classification tasks: audio music mood, audio genre, audio artist and audio tag. In *In Proceedings of the International Symposium on Music Information Retrieval*. Retrieved from http://www.music-ir.org/mirex/abstracts/2008/Peeters_2008_ISMIR_MIREX.pdf

Peeters, G., Giordano, B. L., Susini, P., Misdariis, N., & McAdams, S. (2011). The Timbre Toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, *130*(5), 2902–2916. https://doi.org/10.1121/1.3642604

Pikrakis, A. (2013). *Audio latin music genre classification: A mirex 2013 submission based on a deep learning approach to rhythm modelling*. Citeseer. Retrieved from https://www.music-ir.org/mirex/abstracts/2013/AP1.pdf

Platt, J., Cristianini, N., & Shawe-Taylor, J. (2000). *Large Margin DAGs for Multiclass Classification*. *Advances in Neural Information Processing Systems ,MIT Press* (Vol. 12). https://doi.org/10.1.1.158.4557

Pons, J., & Serra, X. (2018). Randomly weighted CNNs for (music) audio classification. *ArXiv Preprint ArXiv:1805.00237*. Retrieved from http://arxiv.org/abs/1805.00237

Pratt, R. L., & Doak, P. E. (1976). A subjective rating scale for timbre. *Journal of Sound and Vibration*, *45*(3), 317–328. https://doi.org/10.1016/0022-460X(76)90391-6

Provost, F., & Kohavi, R. (1998). Glossary of terms. *Journal of Machine Learning*, *30*(2–3), 271–274.

Ren, J.-M., Wu, M.-J., & Jang, J.-S. R. (2015). Automatic music mood classification based on timbre and modulation features. *IEEE Transactions on Affective Computing*, *6*(3), 236–246. https://doi.org/10.1109/TAFFC.2015.2427836

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*(6), 1161.

Saari, P. (2009). *Feature Selection for Classification of Music According to Expressed Emotion*. University of Jyväskylä. Retrieved from https://jyx.jyu.fi/handle/123456789/22757

Sällberg, B., Grbić, N., & Claesson, I. (2007). Online maximization of subband kurtosis for blind adaptive beamforming in realtime speech extraction. In *2007 15th International Conference on Digital Signal Processing, DSP 2007* (pp. 603–606). https://doi.org/10.1109/ICDSP.2007.4288654

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, *103*(1), 588–601. https://doi.org/10.1121/1.421129

Scheirer, E., & Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 2, pp. 1331–1334). https://doi.org/10.1109/ICASSP.1997.596192

Scherer, K. R., & Oshinsky, J. S. (1977). Cue utilization in emotion attribution from auditory stimuli. *Motivation and Emotion*, *1*(4), 331–346. https://doi.org/10.1007/BF00992539

Schlosberg, H. (1954). Three dimensions of emotion. *Psychological Review*, *61*(2), 81.

Seo, J. S., & Lee, S. (2011). Higher-order moments for musical genre classification. *Signal Processing*, *91*(8), 2154–2157. https://doi.org/10.1016/j.sigpro.2011.03.019

Seyerlehner, K., & Schedl, M. (2014). *Mirex 2014: Optimizing the fluctuation pattern extraction process*. Retrieved from https://www.music-ir.org/mirex/abstracts/2014/SS6.pdf

Seyerlehner, K., Schedl, M., Pohle, T., & Knees, P. (2010). Using block-level features for genre classification, tag classification and music similarity estimation. *Submission to Audio Music Similarity and Retrieval Task of MIREX, 2010*. Retrieved from https://www.music-ir.org/mirex/abstracts/2010/SSPK2.pdf

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Silla Jr, C. N., Koerich, A. L., & Kaestner, C. A. A. (2008). The Latin Music Database. In *ISMIR 2008 : proceedings of the 9th International Conference of Music Information Retrieval* (pp. 451–456). Retrieved from https://books.google.fi/books?hl=en&lr=&id=OHp3sRnZD-oC&oi=fnd&pg=PA451&dq=The+Latin+Music+Database.&ots=oFNKpIdCd8&sig=WoChAqaWn1NnK9uMYkJNBPApmO8&redir_esc=y#v=onepage&q=The Latin Music Database.&f=false

Sloboda, J. A., & O'Neill, S. A. (2001). Emotions in everyday listening to music. *Music and Emotion: Theory and Research*, 415–429. https://doi.org/10.1016/j.physa.2016.05.007

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, & Bernhard Schölkopf. (2006). Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research, 7*(Jul), 1531--1565. Retrieved from http://www.fml.tuebingen.mpg.de/raetsch/projects/shogun.%5Cnhttp://jmlr.csail.mit.edu/papers/volume7/sonnenburg06a/sonnenburg06a.pdf

Sturm, B. L. (2012). An analysis of the GTZAN music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies* (p. 7). ACM. https://doi.org/10.1145/2390848.2390851

Sturm, B. L. (2013a). The gtzan dataset: Its contents, faults, and their effects on music genre recognition evaluation. *IEEE Transactions on Audio, Speech and Language Processing*.

Sturm, B. L. (2013b). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. Retrieved from http://arxiv.org/abs/1306.1461

Sturm, B. L. (2014a). A simple method to determine if a music information retrieval system is a "horse." *IEEE Transactions on Multimedia*, *16*(6), 1636–1644. https://doi.org/10.1109/TMM.2014.2330697

Sturm, B. L. (2014b). The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval. *Journal of New Music Research*, *43*(2), 929–8215. https://doi.org/10.1080/09298215.2014.894533

Thayer, R. E. (1990). *The biopsychology of mood and arousal*. Oxford University Press.

timbre | Origin and meaning of timbre by Online Etymology Dictionary. (2017). Retrieved May 27, 2017, from https://www.etymonline.com/word/timbre

Toh, A. M., Togneri, R., & Nordholm, S. (2005). Spectral entropy as speech features for speech recognition. In *Postgraduate Electrical Engineering & Computing Symposium* (Vol. 1). Retrieved from https://www.researchgate.net/profile/Sven_Nordholm/publication/247612912_Spectral_entropy_as_speech_features_for_speech_recognition/links/54843daf0cf2e5f7ceaccbb9/Spectral-entropy-as-speech-features-for-speech-recognition.pdf

Tsatsishvili, V. (2011). *Automatic Subgenre Classification of Heavy Metal Music*. University of Jyväskylä.

Tzanetakis, G. (2007). *Marsyas submissions to MIREX 2007*. Retrieved from https://www.music-ir.org/mirex/abstracts/2007/AI_CC_GC_MC_AS_tzanetakis.pdf

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, *10*(5), 293–302. https://doi.org/10.1109/TSA.2002.800560

Van Gestel, T., Suykens, J. A. K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., … Vandewalle, J. (2004). Benchmarking least squares support vector machine classifiers. *Machine Learning*, *54*(1), 5–32.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vignoli, F. (2004). Digital music interaction concepts: A user study. In *Proceedings of the 5th International Society for Music Information Retrieval Conference* (pp. 415–420).

Wang, J.-C., Lo, H.-Y., Jeng, S.-K., & Wang, H.-M. (2010). Mirex 2010: Audio Classification Using Semantic Transformation And Classifier Ensemble. In *Music Information Retrieval Evaluation eXchange (MIREX)* (pp. 2–5).

Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, *15*(2), 70–73. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1161901

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13* (pp. 668–674).

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1–3), 37–52.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Wu, M.-J., & Jang, J.-S. R. (2012). *MIREX 2012 Submissions-Combining Acoustic and Multi-level Visual Features for Music Genre Classification*.

Wu, M.-J., & Jang, J.-S. R. (2013). *MIREX 2013 Submissions for Train/Test Tasks (Draft). MIREX-Music Information Retrieval Evaluation eXchange*. Retrieved from https://www.music-ir.org/mirex/abstracts/2013/JJ1.pdf

Wu, M.-J., & Jang, J.-S. R. (2014). *Confidence-based late Fusion for Music Genre Classification*. Citeseer. Retrieved from https://www.music-ir.org/mirex/abstracts/2014/WJ2.pdf

Wu, M.-J., & Jang, J.-S. R. (2015). Combining acoustic and multilevel visual features for music genre classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *12*(1), 10.

Wundt, W. M. (1907). *Outlines of psychology*. W. Engelmann.

Xu, S., & Gu, Y. (2014). *Music genre classification mirex 2014 submissions*. Retrieved from https://www.music-ir.org/mirex/abstracts/2014/XG2.pdf

Yeh, C., Roebel, A., & Rodet, X. (2010). Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(6), 1116–1126.

Yermeche, Z., Grbic, N., & Claesson, I. (2007). Blind subband beamforming with time-delay constraints for moving source speech enhancement. *IEEE Transactions on Audio, Speech and Language Processing*, *15*(8), 2360–2372. https://doi.org/10.1109/TASL.2007.903309

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems* (pp. 3320–3328). https://doi.org/10.1103/PhysRevB.90.155428

Yu, L., & Liu, H. (2003). *Feature selection for high-dimensional data: a fast correlation-based filter solution. Proceedings of the twentieth international conference on machine learning*. Retrieved from https://www.aaai.org/Papers/ICML/2003/ICML03-111.pdf

Abbreviations

Music Datasets

AF	Artist Filtering
GTZAN	George Tzanetakis
MSD	Million Song Dataset
PandaMood	Renato Panda

Spectral Features

MFCCs	Mel-Frequency Cepstral Coefficients
SB-Entropy	Sub-Band Spectral Entropy
SB-Flux	Sub-Band Spectral Flux
SB-Kurtosis	Sub-Band Spectral Kurtosis
SB-Skewness	Sub-Band Spectral Skewness
SB-ZCR	Sub-Band ZCR

Feature Selection

IG [-]	Information Gain Algorithm
Top 2 [-]	Semi-Manual Selection

Feature Statistical Summaries

$(\boldsymbol{\mu}, \boldsymbol{\sigma})$	Mean and standard deviation values of feature/features
$(\boldsymbol{\mu})$	Mean values of feature/features
$(\boldsymbol{\sigma})$	Standard deviation values of feature/features

Abbreviations

Audio & Digital Signal Processing

dB	Decibel
DFT	Discrete Fourier Transform
FDW	Filtered Dependent Windowing
FFT	Fast Fourier Transform
FT	Fourier Transform
GMM	Gaussian Mixture Model
GSV	Gaussian Super Vector
Hz	Hertz
Khrz	Kilohertz
MLVF	Multi-Level Visual Features
PMSC	Principal Mel – Spectrum Components
RP	Rhythm Pattern
STFT	Short-Time Furrier Transform

Learning Algorithms/Classifiers

DCNN	Deep Convolutional Neural Network
K-NN	K- Nearest Neighbors
LR	Logistic Regression
MLR	Multinomial Logistic Regression
NN	Neural Networks
PFC	Pooled Features Classifier
SVM	Support Vector Machines

Abbreviations

Machine Learning

CA	Classification Accuracy
CV	Cross-Validation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
KFCV	K-Fold Cross-Validation
ReLu	Rectified Linear Unit

Music Information Retrieval

AGC	Automatic Genre Classification
AGT	American Annotator Ground Truth (K-Pop)
AMC	Automatic Mood Classification
AMM	Automatic Music Mood
IMIRSEL	The International Music Information Retrieval Systems Evaluation Laboratory
KGT	Korean Annotator Ground Truth (K-Pop)
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation Exchange

Other

ANOVA	Analysis of Variance
DCT	Discrete Cosine Transform
HSD	Honestly Significant Difference
KETI	Korea Electronics Technology Institute
LOG	Logarithm
MIDI	Musical Instrument Digital Interface
PDF	Probability Density Function
PMF	Probability Mass Function

Appendix A

All Classification Stage Accuracies

In this appendix we provide all classification accuracies scores that resulted from our factorial design. In total, we trained and evaluated 117 classification models, 78 of which pertained music genre (GTZAN) and 39 music mood (PandaMood). The following tables are organized on a per task (music genre, music mood) and per feature set basis. The reporting format contains training, testing and artist filter testing (AF.Testing) average classification accuracies and standard deviations of cross-validation (CV). Highlighted in bold are the highest testing accuracies and the lowest training set accuracies.

All tables report the classification results from CV, table A.1 focuses on the ‘All Features’ set, table A.2 on the individual feature set and table A.3 on automatic feature selection (information gain). Tables A.4 and A.5 focus on semi-manual selection (Top 2) with A.4 for GTZAN and A.5 for PandaMood.

Music Genre Classification Accuracies (GTZAN)				Music Mood Classification Accuracies (PandaMood)	
Feature Set SB-Flux (μ, σ)					
Algorithms	**Training Set**	**Testing Set**	**AF.Testing Set**	**Training Set**	**Testing Set**
SVM	71.84%(0.45%)	61.10% (3.04%)	39.76% (3.24%)	53.76% (0.75%)	**36.29%(2.91%)**
MLR	70.69%(0.40%)	**63.99%(4.33%)**	**48.84%(5.70%)**	**40.42%(1.24%)**	35.18%(4.34%)
K-NN	**69.72%(0.74%)**	56.26%(2.93%)	36.10%(0.85%)	53.17%(0.72%)	30.03%(4.65%)
Feature Set SB-Entropy (μ, σ)					
Algorithms	**Training Set**	**Testing Set**	**AF.Testing Set**	**Training Set**	**Testing Set**
SVM	79.59%(0.45%)	66.36%(2.72%)	48.51%(4.70%)	60.16%(0.82%)	**39.51%(4.19%)**
MLR	**72.82%(0.69%)**	**66.77%(5.20%)**	**53.05%(5.49%)**	**44.01%(1.19%)**	38.82%(4.51%)
K-NN	73.98%(0.66%)	61.13%(4.31%)	40.09%(0.27%)	53.99%(1.14%)	34.01%(5.48%)
Feature Set SB-Kurtosis (μ, σ)					
Algorithm	**Training Set**	**Testing Set**	**AF.Testing Set**	**Training Set**	**Testing Set**
SVM	71.96%(0.70%)	59.35% (4.47%)	40.10%(5.80%)	48.79%(0.49%)	36.06%(4.34%)
MLR	**68.26%(0.67%)**	**61.81%(5.72%)**	**50.17%(5.04%)**	**42.96%(0.87%)**	**39.05%(5.92%)**
K-NN	69.03%(0.92%)	52.80% (5.39%)	35.55%(3.03%)	52.02%(0.81%)	29.55%(4.67%)
Feature Set MFCCs (μ, σ)					
Algorithms	**Training Set**	**Testing Set**	**AF.Testing Set**	**Training Set**	**Testing Set**
SVM	80.74%(0.71%)	**60.97%(5.63%)**	40.20%(4.36%)	65.24%(0.67%)	**38.65%(4.35%)**
MLR	**67.60%(0.86%)**	58.57%(4.54%)	**42.97%(3.15%)**	**43.41%(0.84%)**	34.97%(4.38%)
K-NN	71.87%(0.51%)	56.76%(4.63%)	39.09%(2.15%)	54.34%(0.86%)	28.56%(4.40%)
Feature Set SB-ZCR (μ, σ)					
Algorithms	**Training Set**	**Testing Set**	**AF.Testing Set**	**Training Set**	**Testing Set**
SVM	69.54%(0.83%)	54.78%(5.49%)	37.65%(2.26%)	54.19%(1.04%)	**35.40%(4.74%)**
MLR	**62.31%(0.85%)**	**56.00%(6.27%)**	**39.87%(5.14%)**	**39.53%(0.65%)**	32.52%(5.20%)
K-NN	65.55%(0.51%)	49.43%(3.20%)	32.78%(0.70%)	49.98%(0.93%)	26.33%(3.07%)
Feature Set SB-Skewness (μ, σ)					
Algorithms	**Training Set**	**Testing Set**	**AF.Testing Set**	**Training Set**	**Testing Set**
SVM	76.23%(0.67%)	**61.45%(4.04%)**	41.86%(2.48%)	52.43%(0.36%)	37.52%(5.41%**)**
MLR	67.44%(0.87%)	59.90%(4.38%)	**47.62%(2.05%)**	43.02%(0.62%**)**	**37.63%(5.04%)**
K-NN	69.79%(0.76%)	54.33%(5.39%)	37.99%(1.92%)	53.29%(0.58%)	31.65%(4.59%)

TABLE A.2: CV average classification accuracies and standard deviations (in parentheses) of all classifiers and learning tasks for the 'Individual Features' sets.

Music Genre Classification Accuracies (GTZAN)				Music Mood Classification Accuracies (PandaMood)	
Feature Set All Features (μ, σ)					
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**	**Training CA**	**Testing CA**
SVM	92.22%(0.45%)	76.64%(1.42%)	52.83%(5.71%)	70.20%(0.69%)	**42.41%(5.15%)**
MLR	99.00%(0.26%)	**77.83%(4.57%)**	**64.57%(5.72%)**	64.98%(0.84%)	38.73%(4.90%)
K-NN	**80.41%(0.51%)**	71.66%(4.88%)	44.41%(3.48%)	**53.32%(0.91%)**	32.78%(5.07%)
Feature Set All Features (μ)					
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**	**Training CA**	**Testing CA**
SVM	87.61%(0.47%)	71.86%(4.15%)	47.18%(6.48%)	62.88%(1.08%)	**37.76%(3.88%)**
MLR	89.25%(0.58%)	**73.92%(2.98%)**	**58.26%(4.94%)**	**50.49%(0.97%)**	36.85%(5.92%)
K-NN	**77.41%(0.56%)**	64.96%(4.32%)	43.19%(4.26%)	53.60%(0.88%)	31.31%(3.21%)
Feature Set All Features (σ)					
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**	**Training CA**	**Testing CA**
SVM	89.05%(0.30%)	73.16%(3.29%)	51.61%(4.93%)	65.80%(1.12%)	**40.08%(3.38%)**
MLR	90.96%(0.48%)	**73.39%(3.46%)**	**60.25%(2.06%)**	**52.46%(1.01%)**	38.45%(3.59%)
K-NN	**78.60%(0.70%)**	66.28%(4.79%)	44.52%(2.26%)	53.20%(1.16%)	31.75%(4.27%)

TABLE A.1: CV average classification accuracies and standard deviations (in parentheses) of all classifiers and learning tasks for the 'All Features' sets.

Music Genre Classification Accuracies (GTZAN)				Music Mood Classification Accuracies (PandaMood)	
Feature Set Information Gain FS (Top 20)					
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**	**Training CA**	**Testing CA**
SVM	73.22%(0.44%)	64.85%(3.52%)	44.52%(2.26%)	51.58%(0.65%)	**38.73%(3.62%)**
MLR	**72.65%(0.73%)**	**65.53%(5.42%)**	**48.51%(2.93%)**	**43.40%(0.67%)**	37.94%(4.77%)
K-NN	74.06%(0.80%)	63.13%(3.56%)	41.75%(3.92%)	53.11%(0.60%)	33.45%(3.10%)

TABLE A.3: CV average classification accuracies and standard deviations (in parentheses) of all classifiers and learning tasks for the 'Information Gain' automatic feature selection set.

.

Music Mood Classification Accuracies (PandaMood)		
Feature Set Top 2 [SB-Entropy(μ, σ)- MFCCs (μ, σ)]		
Algorithms	**Training CA**	**Testing CA**
SVM	70.06%(0.59%)	**42.18%(2.89%)**
MLR	**50.45%(1.06%)**	39.11%(5.23%)
K-NN	54.79%(0.96%)	32.99%(4.07%)
Feature Set Top 2 [SB-Entropy(μ)- MFCCs (σ)]		
Algorithms	**Training CA**	**Testing CA**
SVM	59.37%(0.96%)	**39.84%(5.42%)**
MLR	**44.53%(0.34%)**	38.87%(2.13%)
K-NN	52.22%(1.02%)	29.14%(4.32%)
Feature Set Top 2 [SB-Entropy(σ)- MFCCs(μ)]		
Algorithms	**Training CA**	**Testing CA**
SVM	66.91%(0.66%)	**38.87%(4.55%)**
MLR	**41.92%(0.97%)**	35.30%(3.55%)
K-NN	52.71%(1.45%)	30.10%(4.56%)

TABLE A.4: CV average classification accuracies and standard deviations (in parentheses) of all classifiers for the 'Top 2' semi-manual feature selection sets in the music mood task.

Music Genre Classification Accuracies (GTZAN)			
Feature Set Top 2 [SB-Entropy(μ, σ)- SB-Flux(μ, σ)]			
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**
SVM	84.69%(0.63%)	70.34%(2.79%)	50.72%(3.82%)
MLR	85.22%(0.56%)	**74.02%(3.68%)**	**58.81%(4.83%)**
K-NN	**77.32%(0.58%)**	66.88%(4.17%)	43.74%(1.05%)
Feature Set Top 2 [SB-Entropy(μ)- SB-Flux(σ)]			
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**
SVM	78.42%(0.69%)	**65.37%(3.93%)**	47.07%(3.49%)
MLR	**72.34%(0.69%)**	65.23%(3.65%)	**53.05%(4.16%)**
K-NN	74.95%(1.05%)	62.53%(5.77%)	40.98%(0.93%)
Feature Set Top 2 [SB-Entropy(σ)- SB-Flux(μ)]			
Algorithms	**Training CA**	**Testing CA**	**AF.Testing CA**
SVM	77.47%(0.62%)	**65.00%(2.32%)**	49.62%(4.26%)
MLR	**71.55%(0.62%)**	64.93%(3.76%)	**49.73%(4.15%)**
K-NN	73.61%(0.62%)	59.99%(3.66%)	40.76%(2.26%)

TABLE A.5: CV average classification accuracies and standard deviations (in parentheses) of all classifiers for the 'Top 2' semi-manual feature selection sets in the music genre task.

Appendix B

Feature Extraction Code

The MATLAB code for extracting each FDW sub-band feature will become available in MIRtoolbox 1.7.2 (Lartillot, Toiviainen, et al., 2008).