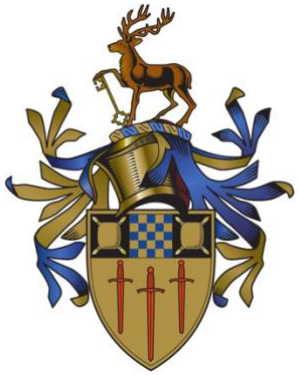
Music Genre Classification using Metaphor-based Metaheuristic Algorithms

Josh Prewer

Institute of Sound Recording
University of Surrey



March 2019

Abstract

Automatic music genre classification alleviates the need for manually listening to and labelling large quantities of music. Current music genre classification methods involve extracting large amounts of features from audio samples to train a machine learning model. Feature selection algorithms aim to reduce the features down to the most relevant descriptors of genre to build a more robust model. The aim of this study is to explore if metaphor-based metaheuristic algorithms can be used for this task. Self-Adaptive Harmony Search (SAHS) is identified as a metaheuristic that has seen successful use in music genre classification. Binary Cuckoo Search (BCS) and Binary Dragonfly Algorithm (BDFA) are identified as further metaheuristics that have seen improved performance over Harmony Search in other classification problems.

An experiment was devised to compare the SAHS, BCS and BDFA algorithms against traditional feature selection algorithms Principal Component Analysis (PCA), ReliefF Sequential Forward Search (ReliefF-SFS) and using no feature selection. A further experiment was conducted comparing no feature selection and using SAHS, BCS and BDFA to create individual feature sets for each pairwise combination of genres. The experiments were conducted using two music genre datasets. The first experiment a showed small possible improvement in performance using metaheuristics in one dataset but a statically significant decrease in performance using the other dataset. In the second experiment, BCS and BDFA showed an insignificant increase in performance on one dataset compared to no feature selection. However, all metaheuristics showed a small but significant drop in performance using the other dataset. Of the three metaheuristics tested, BCS and BDFA outperformed the SAHS feature set but not always by statistically significant results. Increasing the number of iterations of SAHS, BCS and BDFA is suggested for potentially improving the performance.

Acknowledgements

I would like to thank Enzo De Sena and Matt Vowels for their support and advice. I would like to extend this thanks to the rest of the staff at the IOSR for all their hard work and support that has guided me through the past four years. I would also like to thank my housemates and my family for proof reading and the much-needed support. Thanks to Spotify and Dua Lipa for soundtracking a significant period of my life.

Table of Contents

List of Figures

List of Tables

List of Acronyms

TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
OVO	One versus one
OVA	One versus all
GMM	Gaussian Mixture Model
SVM	Support Vector Machine
RBF	Radial Basis Function
STFT	Short Time Frequency Transform
MMFC	Mel Frequency Cepstral Coefficient
SSD	Statistical Spectrum Descriptor
PCA	Principal Component Analysis
SFS	Sequential Forward Search
(SA)HS	(Self-Adaptive) Harmony Search
(B)CS	(Binary) Cuckoo Search
(B)DFA	(Binary) Dragonfly Algorithm

Chapter 1
Introduction

The rise of music streaming and large music databases has increased the use of automatic music genre classification. Genre labelled music allows easier searching within large music databases and can be used to implement recommendation systems. Automatic music genre classification aims to alleviates the task of manually listening to and labelling large amounts of music. Machine learning can be used to identify the patterns that form genre in order to automatically detect a piece of music's genre. These systems involve extracting audio and musical features from various genre labelled audio files. These features are given to a machine learning algorithm which builds a statistical model to determine how the features define genre. Genre predictions can then be made by extracting the same features used to train the system and mapping them onto the statistical model. The features extracted can be evaluated and filtered to provide the most relevant descriptors of genre. Ideally, large amounts of features could be extracted and given to a system, with the most relevant features selected to build a robust model of genre.

Algorithms used to evaluate and select these relevant features are known as feature selection algorithms. Recently, a new set of algorithms known as metaphor-based metaheuristics have been applied to features selection. These are problem independent algorithms used for solving optimisation problems, based on natural or man-made metaphors. This paper aims to answer the following question; how can metaphor-based feature selection algorithms be successfully implemented in music genre classification?

In order to answer this question, this work investigates the following:

- What machine learning principles and algorithms are used to classify genre?
- What are the extractable features that can be used to identify genre?
- What feature selection algorithms can be used to find relevant features?

A literature review is conducted in Chapter 2, Chapter 3 and Chapter 4. Chapter 2 discusses the machine learning principles and algorithms used to classify genre. Chapter 3 assesses the features that can be extracted to predict genre. Chapter 4 looks into feature selection algorithms and whether metaphor-based metaheuristic algorithms can be used for this task. Metaheuristics that have seen use in music

genre classification are identified, along with algorithms that have seen success in other classification problems.

Chapter 5 summarises the findings of the literature review and explores the gaps in current research. Further research questions are discussed leading to some experimental hypotheses. An experiment testing these hypotheses is devised in Chapter 6. The subsequent results are analysed and discussed in Chapter 7. Finally, Chapter 8 makes conclusions based on the results and proposes further work.

Chapter 2
Machine Learning Principles

Machine learning is the field of computing and statistics where data is utilised to perform tasks without explicit instructions. Music genre classification systems utilise machine learning to identify trends characterising genre. These trends can be used to generate predictions on unknown data. This chapter will look at the principles of machine learning and how they can be applied to music genre classification. In this chapter the following questions will be answered:

1. How is machine learning used in music genre classification?
2. What are the best methods for analysing the performance of machine learning classification systems?
3. What classification algorithms are used in music genre classification?

2.1. General Principles

Machine learning is defined as a set of methods that can automatically detect patterns in data and use these uncovered patterns to predict future data [Murphy, 2012]. Computational learning algorithms are used to generate models based on provided data to perform a task. This is useful for large datasets where the patterns are not obvious and require complex analysis to discover them [Alpaydin and Bach, 2014].

Machine learning algorithms can be grouped into two types: supervised and unsupervised learning. Supervised learning involves building a statistical model from a set of data. This model is used for predicting or estimating an output based on one or more inputs. This type of learning is useful for predicting trends in data and identifying associated causes. In unsupervised learning, a set of data containing only inputs is used to find unknown trends in the data. Work exists on unsupervised learning in music genre classification [Barreira et al., 2011; Shao et al., 2004]. However, recent success and extensive research on supervised learning methods makes it of interest in this paper.

In supervised learning, labels are defined as the values to be predicted. Features are the input variables used in the system. A model is trained using labelled features to establish a relationship between the labels and features. The labels can either be quantitative or qualitative values. Predicting quantitative value labels is referred to as regression problems. Predicting qualitative value labels is referred to as classification problems. Music genre classification can be treated as a classification problem. This can

be solved using supervised learning methods, provided that datasets of genre labelled music exist. Datasets for classification can either be balanced or unbalanced. Balanced datasets have an equal number of samples for each possible class. Unbalanced datasets have an unequal number of samples for each possible class [Bowles, 2015]. GTZAN (balanced) and ISMIR2004 (unbalanced) are two datasets used in music genre classification.

2.2. Classification

Identifying the music genre of an audio file is a classification problem due to the qualitive nature of the output of the model. Music genre can be rock, pop, classical, blues etc, all categorical values. Binary classification refers to a classification problem of two classes. If the number of classes is greater than two it is known as multiclass classification [Murphy, 2012]. For multiclass classification, multiple instances of a single binary classifier are used to generate a model that can distinguish between the multiple classes. These instances can be combined either using a one versus one or one versus all approach.

2.2.1. One Versus All

A one versus all (OVA) approach consists of fitting one binary classifier per class. Classifiers are trained using the positive samples of one class, with the remaining samples being negative. In the context of music genre classification, this would result in classifiers identifying if a track is or isn't rock, or if a track is or isn't reggae. This method suffers from the training sets of the binary classifiers being imbalanced. If a dataset contains balanced samples of ten classes, an individual classifier will be trained on 90% negative samples and only 10% positive samples [Bishop, 2006]. Furthermore, if the original dataset is imbalanced, the confidence level between individual binary classifiers will vary. Further ambiguity can occur when the classifiers are combined if they received the same number of votes for classifying different genres. In both the one versus one and one versus all approaches, final classification can be achieved using a majority voting scheme between all the binary classifiers. Majority voting takes into account the most likely prediction as well as the confidence level to handle ambiguous cases.

2.2.2. One Versus One

A one versus one (OVO) approach consists of fitting a binary classifier per class pairing, resulting in the total number of classifiers being trained as:

$$\boldsymbol{number\ of\ classes} \times \frac{\boldsymbol{number\ of\ classes - 1}}{\mathbf{2}} \quad \textbf{(1)}$$

In the context of music genre classification, this would result in classifiers identifying if music is rock or pop, or if music is reggae or blues. Provided the original dataset is balanced, this resolves the

imbalanced dataset issue of one versus all but requires more computation. However, it still suffers from the different confidence levels of binary classifiers for imbalanced datasets and potential ambiguity when the binary classifiers are combined.

2.2.3. Performance Metrics

Accuracy is one metric used for evaluating classification models and is defined as:

$$\boldsymbol{Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}} \quad \textbf{(2)}$$

The accuracy metric can be applied to both the training set and test set of a dataset, but a high test set accuracy is what makes a *good* classifier [Dietterich and Bakiri, 1995]. For class imbalanced datasets, accuracy isn't a descriptive metric. For example, for a dataset of tumours, 91 are benign and 9 are malignant. If the model is 91% accurate, only 1 out of nine malignant tumours will be detected. Precision and recall are another metric used in model evaluation. For classification problems, the response could be one of four outcomes:

- **True positive (TP)**: the response is correctly classified as the positive label. *Rock music is classified as rock music.*
- **False positive (FP):** the response is classified as the negative label when it is a positive label. *Rock music is classified as not rock music.*
- **True negative (TN):** the response is correctly classifier as the negative label. *Not rock music is classified as not rock music*.
- **False negative (FN):** the response is classified as the positive label when it is the negative label. *Not rock music is classified as rock music.*

Precision describes the number of positive predictions that were actually correct and is defined as:

$$\boldsymbol{Precision = \frac{Number\ of\ TP}{Number\ of\ TP\ and\ FN}} \quad \textbf{(3)}$$

Recall describes the number of positive samples that were correctly identified and is defined as:

$$\boldsymbol{Recall = \frac{Number\ of\ TP}{Number\ of\ TP\ and\ FP}} \quad \textbf{(4)}$$

These metrics better evaluate an imbalanced classification model [Alpaydin and Bach, 2014]. These results can further be combined to produce an F1 score which describes the harmonic mean of precision and recall. It gives a broader sense of overall performance than either one individually and is defined as:

$$F_1 = 2 \cdot \frac{\mathbf{precision} \cdot \mathbf{recall}}{\mathbf{precision} + \mathbf{recall}} \quad (5)$$

When minimising the error of a system, a trade-off exists between bias and variance. Variance refers to the degree that the output will change when the model is trained with a different dataset. Bias represents the predictable degree to which all of the output is inaccurate [Alpaydin and Bach, 2014]. Bias is an indication of constant error in the dataset or a model approximating a real-life problem too simply. Training a flexible system may perfectly represent the dataset and exhibit low bias. However, due to inevitable variance in data, this model will no longer represent the greater population. Predictions with this model will exhibit high variance. Patterns learnt by the model that improve the performance on the training data but not the test data are known as overfitting the data. Conversely, training an overly simple model will produce similar results irrespective of the dataset used and exhibit high bias. This is known as underfitting [Shalev-Shwartz and Ben-David, 2014].

To avoid overfitting, machine learning experiments split the available data into a training and test set. When evaluating different settings for classifiers, there is still the risk of overfitting. These settings can be tweaked until the algorithm performs optimally on the test set of data. This results in evaluation metrics that no longer report on the performance of the algorithm. Another set of the data can be held out to alleviate this, known as a validation set. The training set is used to train a model, followed by an evaluation stage using the validation set. If the evaluation stage is successful, final testing is achieved using the test set. This method reduces the amount of available data that can be used to train the model. However, the evaluation performance can be highly variable, depending on how the training, validation and test data is split [James, 2013].

Cross validation is designed to give an accurate estimate of true error without wasting too much data. The basic approach, k-fold cross validation, splits a training set into k subsets (folds). For each of the folds, a model is trained using a union of the other folds and is evaluated using the current fold. The performance measure is the averaged evaluation value of each of the folds. This method can be computationally expensive but wastes little data [Shalev-Shwartz and Ben-David, 2014].

There is a bias-variance trade off associated with the choice of k in k-fold cross validation. A k of 5 or 10 produces test error rate estimates that are neither excessively high in bias nor variance [James, 2013].

2.3. Classification Algorithms

This section presents different classification algorithms used in music genre classification. Some classification algorithms consist of hyperparameters. These are parameters which are set before the learning process begins and can affect the performance of the model. Hyperparameters are often tuned using either grid search or random search. Grid search builds a model for every combination of

hyperparameters specified. Random search builds a model from randomly selected hyperparameters [Bergstra and Bengio, 2012].

2.3.1. Gaussian Mixture Models

A Gaussian mixture model (GMM) is a probabilistic model that is generated from the sum of different gaussian distributions. It aims to estimate a probability density function (pdf) for the feature values of a class. Each class is assumed to consist of a mixture of a number of multidimensional Gaussian distributions. This mixture may give rise to complex pdfs and can be expressed as:

$$\boldsymbol{p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|u_k, \Sigma_k)} \quad \textbf{(6)}$$

where K is the number of Gaussian distributions (components) and $\mathcal{N}(x|u_k, \Sigma_k)$ is the individual Gaussian density with a mean u_k and covariance Σ_k. The parameter π_k is known as the mixing coefficient and scales each Gaussian in the mixture. A GMM is trained by optimising the parameters μ_k, Σ_k and π_k using the iterative expectation-maximization algorithm (EM). This algorithm alternates between an expectation step (E) and a maximisation step (M). The expectation step finds the log-likelihood of the current estimate for the parameters. The maximisation step alters these parameters to maximise the expected log-likelihood. These parameters are then used in the next expectation step. The number of components K is a common hyperparameter available in GMM implementations.

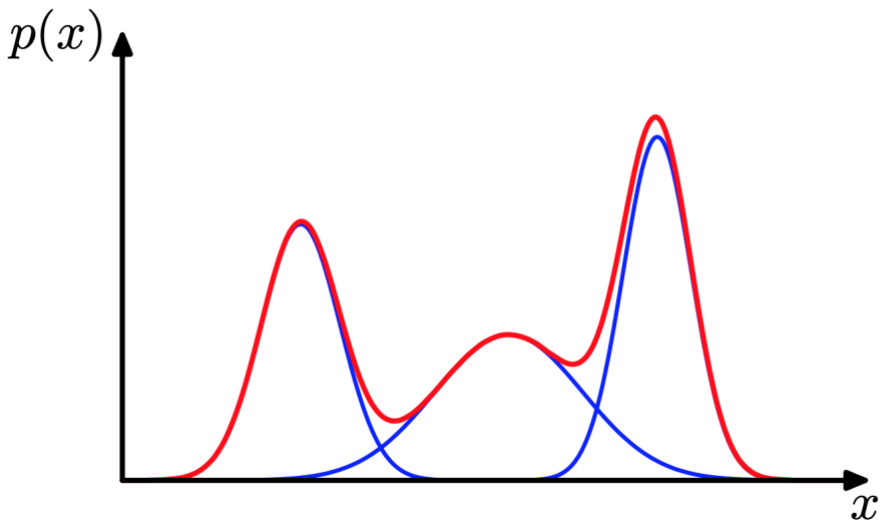


Fig 1: A three-component Gaussian mixture distribution. Components are shown in blue and the summed pdf is shown in red. Originally from [Bishop, 2006].

Tzanetakis and Cook [2002] first introduced music genre classification using machine learning. Their research compared a Gaussian mixture model against a k-nearest neighbour algorithm. This is another

machine learning algorithm which consists of mapping data to a feature space. Predictions are made based on the nearest features to the where the input features fall in the space. The dataset consisted of a mixture of pitch, beat, timbre and spectral features equating to 30 different variables. The GTZAN genre dataset was used to train and test the data using ten-fold cross validation. A three component GMM classifier was found to have a performance of 61% accuracy.

A more recent study from Holzapfel and Stylianou [2007] continued the research of GMM classifiers in genre classification. The feature set used was based on a non-negative matrix factorisation approach, where a spectrogram is decomposed into two separate matrices representing the spectral base of a signal and its respected weights. A classification accuracy of 72.9% was observed when tested on the GTZAN dataset and 70.8% on the ISMIR2004 dataset using 5-fold cross validation. GMM classifiers have seen declining use in literature with recent studies employing support vector machines instead [Baniya and Lee, 2016; Huang et al., 2014; Rosner and Kostek, 2018].

2.3.2. Support Vector Machines

Support vector machines (SVM) are supervised learning algorithms used in classification and regression. An SVM model maps data into a multi-dimensional space where hyperplanes divide the space into different classes. These hyperplanes are mapped to obtain the maximum margin between data (shown in Fig 2). New data samples can be mapped to the space and class predictions made dependent on where it falls in relation to the hyperplanes. A soft margin is employed to construct hyperplanes that allow data points to be crossed. A tuning parameter C is used to govern the severity of which these data points can cross the hyperplane. A high C can be employed to reduce the errors caused by bias. Conversely, a low C can be used to reduce the errors due to variance. Kernel functions can be applied to SVM algorithms to learn nonlinear classification rules by altering the hyperplanes to be nonlinear. The traditional linear kernel can be replaced with a Radial Basis Function (RBF) or a Polynomial kernel. The gamma hyperparameter is used when the kernel isn't linear. Gamma defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. The basic SVM classifier supports only binary classification. Multiclass classification can be achieved using binary SVM classifiers and the aforementioned techniques in 2.2 [Bishop, 2006].

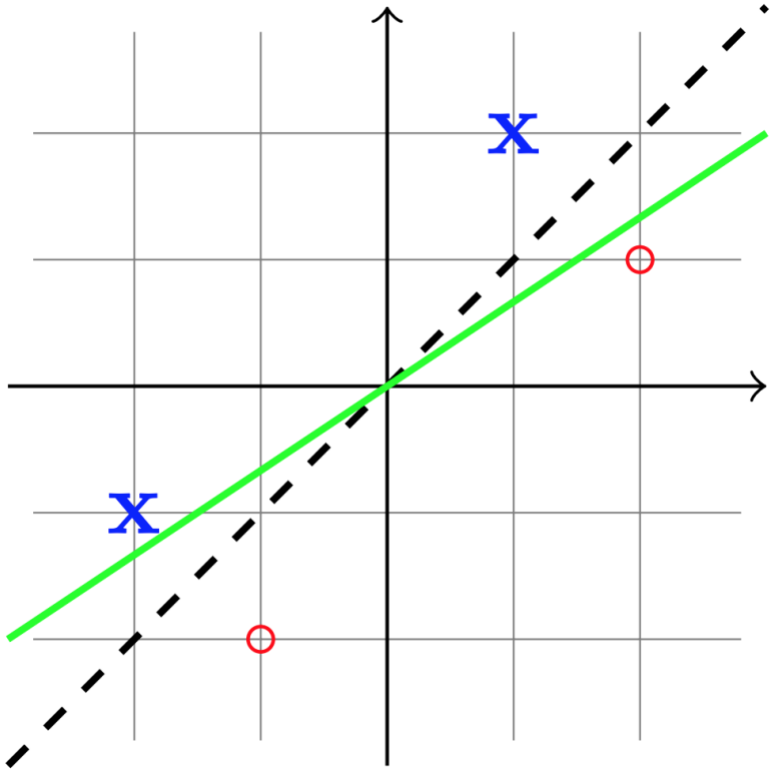


Fig 2: A plot of data showing potential hyperplanes. SVM will use the blacked dashed hyperplane to classify data due to its maximal margin separation. Originally from [Shalev-Shwartz and Ben-David, 2014].

SVM's have seen extensive use in music genre classification. Following research on using Gaussian mixture models, Li and Tzanetakis [2003] presented the same experiment used in [Tzanetakis and Cook, 2002] but evaluated the use of SVM. The results of the paper showed an improvement on the previous research, with the SVM classifier respectively scoring 69.1% on a reduced feature set compared to 61% of the previous study [Tzanetakis and Cook, 2002]. Further research of genre classification has continued with SVMs. SVMs have continued to be implemented in modern systems with state of the art results reported [Baniya and Lee, 2016; Huang et al., 2014; Rosner and Kostek, 2018].

2.4. Summary

This chapter has discussed the fundamentals of machine learning, correct practices in developing and evaluating a model and the methods used for classification problems generally and within music genre classification.

How is machine learning used in music genre classification?

Machine learning consists of using statistical algorithms to solve a task without explicit instructions. Supervised learning algorithms can be used on datasets containing labelled data to predict the labels of future data. Features are extracted from datasets to build these statistical models. This is suitable for the

task of music genre classification provided large datasets of genre labelled music exists. Datasets can either be balanced or unbalanced. GTZAN and ISMIR2004 are two commonly used genre datasets.

What are the best methods for analysing the performance of machine learning systems?

Machine learning algorithms are susceptible to the datasets used in training. To make the best use of a given dataset but avoiding overfitting, k-fold cross validation can be used to assess the performance of a system. The classification accuracy is a suitable property to analyse provided the dataset in use is balanced. Other performance metrics include precision and recall which can be combined to provide an F1 score which is better suited to imbalanced datasets.

What classification algorithms are used in music genre classification?

Provided a dataset contains quantitively labelled data, machine learning can be used to solve classification problems along with regression problems. One versus one and one versus all methods can be used to combine binary classifiers to solve multiclass problem. Support vector machines and Gaussian mixture models are classification systems that have seen use in music genre classification. Support vector machines have demonstrated optimal performance compared to Gaussian mixture models and continue to be researched. All classification algorithms rely on strongly genre correlated features for accurate predictions.

Chapter 3
Feature Extraction

A problem with classifying genre is how to define it. Different genres place emphasis on different musical features. The literature has commonly broken genre into at least three different feature sets: timbral, rhythmic and tonal features [Chathuranga and Jayaratne, 2013; Huang et al., 2014; Tzanetakis and Cook, 2002]. This chapter will look at the features representing these musical elements and answer the following questions:

1. What timbral features are typically used to classify genre?
2. What rhythmic features are typically used to classify genre?
3. What tonal features are typically used to classify genre?

3.1. Timbral Features

Timbral features have been used in genre classification extensively [Kruspe et al., 2011; Sharma et al., 2018; Tzanetakis and Cook, 2002]. Timbral features represent the perceived sound of a musical note, sound or tone. This allows for two sounds having the same pitch and loudness to be perceived differently. Factors such as instrumentation and playing technique influence timbre. Timbral features are often extracted using a Short Time Fourier Transform (STFT), where the input signal is separated into frames (windowed). Different statistical methods can be used to analyse these frame vectors. The mean and standard deviation of each frame is commonly used, along with the mean and standard deviation of the differences between each frame. Kurtosis and skewness have also been used [Seo and Lee, 2011]. Kurtosis is a measure of whether the data is heavy-tailed or light-tailed relative to a normal distribution. Skewness is a measure of the lack of symmetry within a distribution.

Spectral Features

Spectral features are used to describe the characteristics of a spectrum. Examples of these are the spectral centroid, rolloff and flux.

The spectral centroid is defined as the weighted mean of the frequencies present in a signal and represents the centre of mass of a spectrum. A higher centroid value is linked to a brighter timbre of sound [Schubert et al., 2004]. Mathematically, this is described as:

$$\boldsymbol{C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]}} \quad \textbf{(7)}$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n.

The spectral rolloff describes the spectrums shape and is defined as the frequency below which 85% of the magnitude is concentrated. Mathematically, it is described as:

$$\boldsymbol{\sum_{n}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n]} \quad \textbf{(8)}$$

where R_t is the frequency below which 85% of the magnitude is concentrated.

The spectral flux is defined as the variation value of the spectrum between adjacent frames and measures the rate of change of the spectrum. Mathematically, it is described as:

$$\boldsymbol{F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2} \quad \textbf{(9)}$$

where $M_t[n]$ and $M_{t-1}[n]$ are the normalised magnitudes of the STFT at the current frame t, and the previous frame $t-1$.

These spectral features were first used in [Tzanetakis and Cook, 2002] and have continued to see use in systems [Baniya and Lee, 2016, 2016; Huang et al., 2014].

Mel Frequency Cepstral Coefficients (MFCC)

Mel frequency cepstral coefficients are perceptual features taken from a STFT and used widely in automatic speech recognition and genre classification. They are useful for analysing timbral elements of audio as most of information is contained within the first few coefficients [Tzanetakis and Cook, 2002]. Fig 3 demonstrates the extraction process. They are generated from a STFT of a windowed signal with a series of triangular filters on a Mel scale applied. The Mel scale is perceptual scale of pitches which aims to more accurately reflect human perception of frequency and pitch [Stevens et al., 1937]. A discrete cosine transform (DCT) is applied to the filter banks. This is a method used to represent a signal as a sum of sinusoids at varying magnitudes and frequencies. The transform serves to decorrelate the filter banks and concentrate the spectral information into the first few coefficients.

Tzanetakis and Cook [2002] introduce MFCC's to music genre classification. The first 13 coefficients have been used for speech recognition but the first 5 were found sufficient in genre classification. 10 MFCC features were extracted consisting of the mean and standard deviation of the first 5 coefficients.

These were compared against other feature sets consisting of 5 pitch features, 6 beat features and 9 STFT features. MFCC's outperformed the individual feature sets with a classification accuracy of 47%.

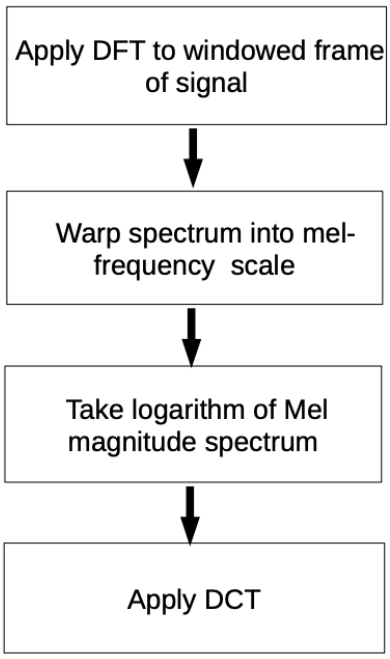


Fig 3: Block diagram of MFCC extraction.

MFCC's have continued to be used in modern genre classification systems with recent studies using the skewness and kurtosis of the coefficients in different frames [Seo and Lee, 2011].

Statistical Spectrum Descriptors (SSD)

Statistical Spectrum Descriptors are seven features derived from a Bark scale spectrogram. The Bark scale is a psychoacoustical scale defined so the critical bands of human hearing each have a width of one Bark. The scale ranges from 1 to 24 to match the first 24 critical bands of hearing [Zwicker, 1961]. Fig 4 demonstrates the extraction process. SSDs are extracted by using an STFT to extract the specific loudness sensation in different critical bands. Spreading functions are applied to account for masking effects of the bands. The spectrum is then converted to the Sone scale to reflect the human loudness sensation (Sonogram). The mean, median, variance, skewness, kurtosis, min and max values of the energy in all 24 critical bands are calculated resulting in a 168 long feature vector.

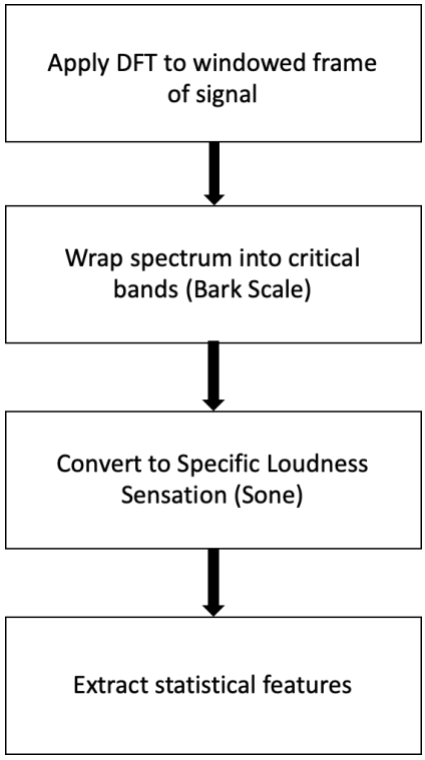


Fig 4: Block diagram of SSD extraction.

Lidy and Rauber [2005] researched the use of SSDs in music genre classification systems. Combined with a support vector machine the feature set scored an accuracy of 72.7% on the GTZAN dataset. Later studies from [Chathuranga and Jayaratne, 2013; Fulzele et al., 2018; Sharma et al., 2018] have included this feature set in their work.

3.2. Rhythmic Features

Genres may have different rhythmic or tempo associations with them. These features aim to represent the associated rhythm of a piece using differing methods.

Rhythm Histogram

Rhythm histograms have been implemented in genre classification and describe the timing of beats within a signal [Tzanetakis and Cook, 2002]. Fig 5 demonstrates the extraction process. A discrete wavelet transform (DWT) is applied to the signal to break it down into octave frequency bands. The time domain amplitude envelope of each band is extracted separately using full wave rectification, lowpass filtering, down sampling and then mean removal. The envelopes of each band are then summed together to find where the signal is most like itself. The first three peaks of output give an indication of tempo. Peak height indicates the strength of a beat. When used as an individual feature vector, the rhythm histograms performed poorly using a GMM classifier achieving 28% accuracy on the GTZAN dataset [Tzanetakis and Cook, 2002].

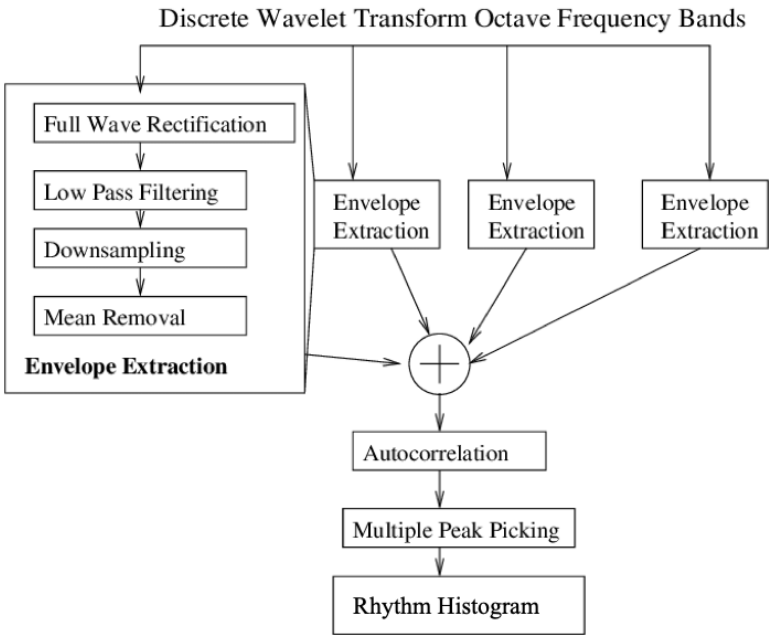


Fig 5: Block diagram of Rhythm Histogram extraction. Adapted from [Tzanetakis and Cook, 2002].

Pattern Clustering

Tsunoo et al. [2011] proposes an alternate method for representation of rhythmic features, separating the rhythm pattern from the bass line pattern. Bar long unit rhythmic patterns are extracted and used to build a pattern occurrence histogram. A harmonic/percussive sound separation (HPSS) algorithm is first applied to isolate the harmonic and percussive elements of the sound. A one pass dynamic programming algorithm (technique used for solving speech recognition problems [Ney, 1984]) is used to extract the unit patterns and overall rhythm map from the percussive signal. The bass line patterns are extracted using the calculated rhythm map and k-means clustering. K-means clustering is an unsupervised learning algorithm used to categorise data into K number of clusters. These feature vectors were tested using a linear SVM classifier on the GTZAN dataset. Using the bass line and rhythm feature vectors individually resulted in 38.2% and 32.7% accuracy. When both were combined with timbral features, this rose to 76.1%.

Rhythm Patterns

Rhythm patterns (also called Fluctuation Patterns) describe the modulation amplitude for a range of modulation frequencies on critical bands of the human auditory range. The extraction process is composed of two stages. The first stage is the same process used to extract SSD features. The second step consists of transforming the spectrum into a time invariant representation based on the modulation frequency of each critical band. This is achieved by applying another discrete Fourier transform, resulting in amplitude modulations of the loudness in individual critical bands [Lidy and Rauber, 2005].

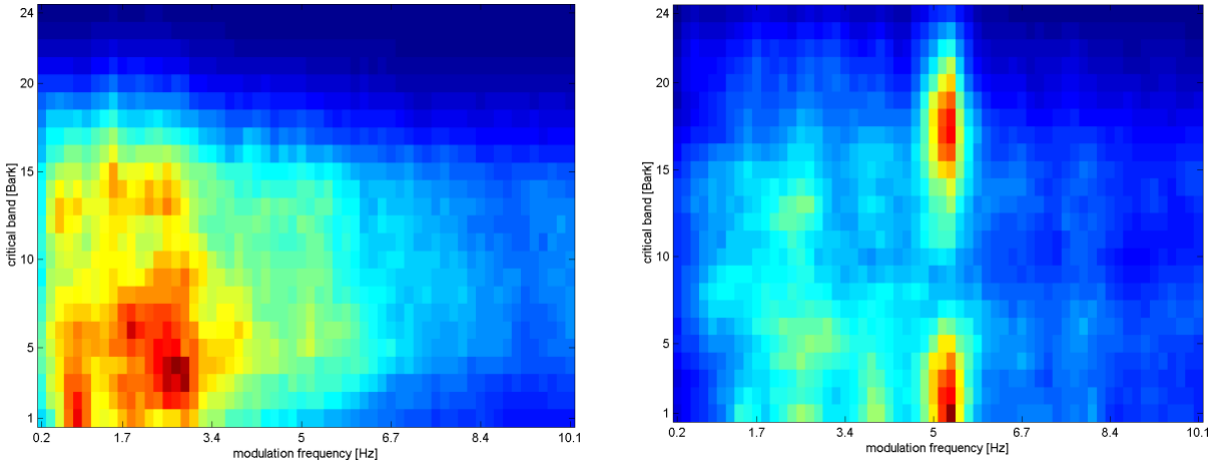


Fig 6: Plot of rhythm patterns for classical (left) and rock (right) music. Originally from [Lidy and Rauber, 2005].

Chathuranga and Jayaratne [2013] included rhythm patterns in their research of music genre classification. Rhythm pattern features were compared against Rhythm histograms and SSD features. It should be noted that this resulted in 1440 individual features for rhythm pattern, 168 for SSD and 60 for rhythm histograms. Fig 6 shows a plot of these 1440 features. An SVM classifier was used to test the GTZAN dataset on each of the feature sets. Rhythm patterns had a reported accuracy of 63.8% while rhythm histograms had an accuracy of 46.4%. The SSD feature set had the best performance with 71.4% accuracy. A simple way to consolidate the feature vector of the rhythm patterns would be to take the mean and standard deviation of each critical band.

3.3. Tonal Features

Tonal features are used to represent the musical content of a piece. These features can identify differences in music associated with key, use of chords, melody and harmony.

Pitch histograms

Tzanetakis and Cook [2002] included pitch features in their work. Features of pitch content are found in a calculated folded pitch histogram and unfolded pitch histogram. The folded histogram contains harmonic content information and the unfolded histogram contains information about the pitch range of the piece. The signal is first divided into two frequency bands, above and below 1 kHz. Both these frequency bands are half-wave rectified and low-pass filtered to extract the amplitude envelopes. The envelopes are then summed. An autocorrelation function is applied to remove the effect of the harmonics of peak frequencies on pitch detection. The prominent peaks of this summary correspond to the main pitches of the signal. The three dominant peaks are accumulated across the whole audio signal into a pitch histogram. The peaks are converted to a musical note using the MIDI note numbering scheme. The unfolded pitch histogram represents the pitches across a number of octaves. For the folded

pitch histogram, all the notes are mapped to a single octave. Five features were taken from these histograms:

- Amplitude of the maximum peak of the folded histogram. This corresponds to the most dominant pitch of the song which will typically be the dominant or tonic note.
- Period of the maximum peak of the unfolded histogram. This corresponds to the octave range of the dominant music pitch
- Period of the maximum peak of the folded histogram. This corresponds to the main pitch class of the song
- Pitch interval between two most prominent peaks of the folded histogram. This corresponds to the main tonal interval relation.
- The overall sum of the histogram. Represents a measure of the strength of the pitch detection

This feature set was tested on the GTZAN dataset using a GMM classifier and achieved an accuracy of 23%.

Chromagram

Chromagrams can be used to extract various tonal features from a signal. A chromagram is a spectrogram that represents the spectral energy of each of the twelve pitch classes by mapping all frequencies to one octave. Chromagrams can be calculated using various methods. The traditional method is to employ a STFT to extract a spectrogram across an audio signal. The intensities of all the frequency bins within the boundaries of a semitone are summed. The semitones in one octave distance are summed to pitch classes resulting in a pitch class profile (PCP) vector. Another method replaces the STFT with a constant-Q transform for time-frequency transformation. The constant-Q transform filters have geometrically spaced centre frequencies and can be dimensioned such that they correspond to musical notes. This allows the chroma features to be computed from the constant-Q transform directly [Stein et al., 2009]. The last method uses the traditional PCP method, but a number of extra steps are taken. Each chroma vector is first normalised and quantised based on log like amplitude threshold. This is known as a CENS (Chroma Energy Normalised) Chromagram and is designed to be more robust to dynamics, timbre and articulation of pitches [Müller and Ewert, 2012]. All the methods can employ windowed analysis, with mean and standard deviation of each chroma being used as features.

Müller et al. [2005] first developed an audio matching system using CENS chromagrams. This was implemented on a classical music database with the goal of finding audio clips that represent a given clip. The example given was to find all interpretations of the theme of Beethoven's Fifth symphony. Matching was successfully achieved using 20 second samples of musical pieces. This problem domain isn't strictly classification, but the impressive results of CENS chromagrams and existing use of chromagrams in genre classification make the CENS implementation of interest.

[Baniya and Lee, 2016; Chathuranga and Jayaratne, 2013; Huang et al., 2014] have included chromagram features in their work with successful results. Although a common feature, no literature exists on comparing chromagrams to other tonal features.

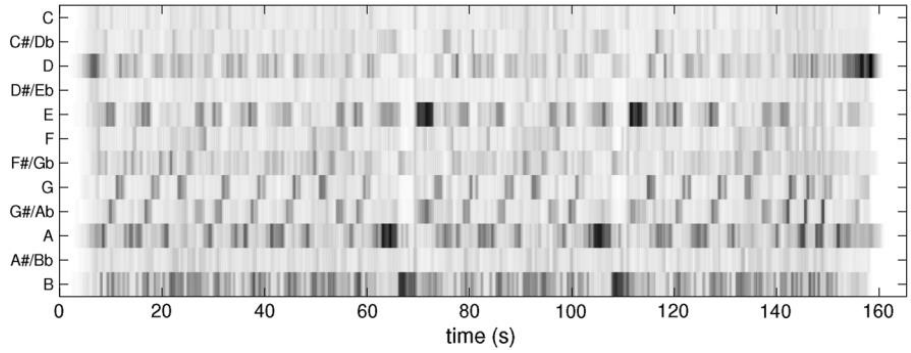


Fig 7: A chromagram extracted from an audio signal using frame based STFT. Originally from [Oudre et al., 2011].

Tonal Centroids

Chromagrams can be used further to extract the tonal centroids of a signal. These represent a 6-dimensional vector corresponding to the projection of the chords along circles of fifths, of minor thirds and of major thirds. This is calculated by multiplying the chroma vector with a transformation matrix. The transformation matrix represents the 6-D space in which a collection of pitches (or a chord) can be described by a single centroid. This allows chords with a tonal centre to lie on a point in the circle of fifths. Chords without a tonal centre will lie in the centre of the circle of fifths [Harte et al., 2006].

The Harmonic Change Detection Function (HCDF) represents the flux of the tonal centroids and can be used to detect chord changes. Fig 8 demonstrates the extraction process. Other changes in harmonic content such as strong melody or bass line movement will be detected [Harte et al., 2006]. [Baniya and Lee, 2016; Huang et al., 2014] both included the HCDF in their feature sets.

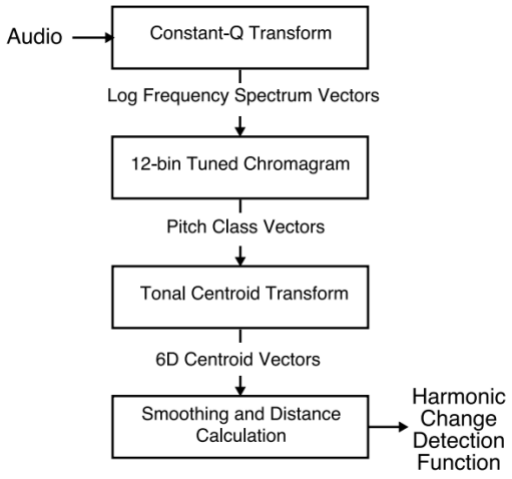


Fig 8: Block diagram of HCDF extraction. Originally from [Harte et al., 2006].

3.4. Summary

This chapter has highlighted some common features used in music genre classification. Classification algorithms rely on strong genre correlated features for successful predictions. Features from a range of musical elements have been presented. However, the strength of correlation of features can vary within these specific feature sets and with comparing differing genres.

What timbral features are typically used to classify genre?

Timbral feature represent the differing characteristics of a sounds with the same pitch and loudness. Different instruments and style of playing cause timbral differences. Spectral features are low level features used to describe the characteristics of spectrum. MFCC's have been shown to be useful timbral features in classifying genre. More recent timbral features have included SSD's which are similar to MFCC's but use the bark scale and consist of more windowed analysis techniques.

What rhythmic features are typically used to classify genre?

Rhythmic features attempt to provide a description of the rhythm in a piece. Basic features such as tempo and metre can be used along with higher level features including Rhythm Histograms, Pattern Clustering and Rhythm Patterns. Rhythm Patterns have shown to provide improved classification results compared to Pattern Clustering and Rhythm Histograms.

What tonal features are typically used to classify genre?

Tonal features are used to provide a description of musical tonality of music. Chromagrams attempt to translate an audio spectrum to a spectrum of musical pitch. From chromagrams, tonal centroids can be extracted to identify harmonic features and changes. Recent literature has employed these chromagram features alongside the tonal centroids and the harmonic change detection function to represent chordal elements of the music.

Chapter 4
Feature Selection Algorithms

This chapter is focused on techniques used to reduce the number of features in a model to better represent the data. Chapter 2 highlighted the need for classification algorithms to have strong genre correlated features. Chapter 3 demonstrated some common features used in music genre classification but highlighted the issue of correlation within these feature sets and differing genres having differing correlated features. Feature selection (FS) is the technique of selecting a subset of these features for use in learning by removing the most redundant and irrelevant features [Kruspe et al., 2011]. Feature selection can provide simpler and more robust models with improved performance. These methods are done during the development process of a model. When predictions are needed to be made the correct features can be pulled out quickly without having to go through the FS algorithm again.

Feature selection algorithms can be split into wrapper methods and filter methods. Filter methods rank the list of features in the training data, regardless of the learning algorithm to be applied. The features are assessed independently of other features according to a quality measure. These methods are effective in computation time and robust to overfitting [Wu et al., 2016]. Wrapper methods require one learning algorithm assess how well the subsets of variables performed. This learning algorithm is often the classification/regression algorithm used for the final model. This allows detection of relationships between features and results in better performance compared to filter methods. However, wrapper methods suffer from being more computationally expensive along with a risk of overfitting if the number of samples in the dataset is insufficient [Alpaydin and Bach, 2014; Chathuranga and Jayaratne, 2013]. A range of filter and wrapper methods will be presented.

This chapter will first look at traditional feature selection algorithms, followed by the use of metaheuristics as feature selection algorithms and answer the following questions:

1. What traditional feature selection algorithms have been used in music genre classification?
2. What metaheuristic feature selection algorithms have been used in music genre classification?
3. What further metaheuristic feature selection algorithms can be used in music genre classification?

4.1. Traditional Feature Selection Algorithms

Traditional feature selection will be PCA and ReliefF will be presented along with their use in music genre classification.

4.1.1. Principal Component Analysis

Principal component analysis is a filter feature selection algorithm that has seen use in music genre classification systems [Panagakis et al., 2010; Yaslan and Cataltepe, 2006]. PCA aims to transform the existing data into fewer dimensions known as principal components. In the context of music genre classification, PCA may find a correlation between the tempo and key in classifying genre. The tempo and key would be then transformed to a single feature (principal component). No assumptions are made about the existence or groupings within the data. PCA can therefore be described as an unsupervised feature selection technique. The transformation is defined such that the first principal component has the largest variance with the variance decreasing in subsequent components. Therefore, the first few principles will account for the most significant representation of the data. PCA implementations can control the number of components to be used or the minimum variance required for components.

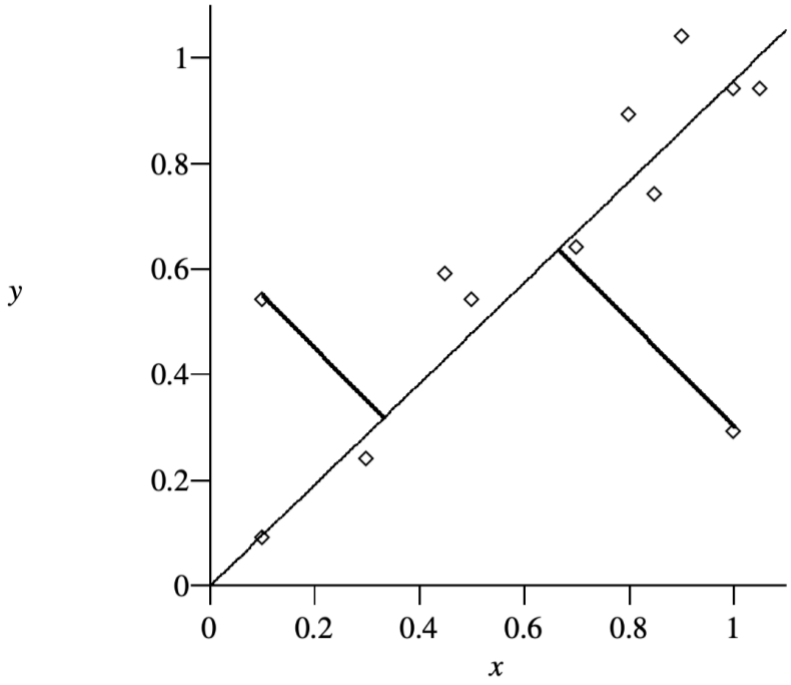


Fig 9: Principal components line of best fit. Originally from [Webb, 2003].

In Fig 9, x and y represent features with each point representing that feature value of a sample. Two lines of best fit can be plotted to this chart, dependent on if x is the independent variable ($y = mx + c$) or y is the independent variable ($x = my + c$). PCA produces a single line of best fit determined by the sum of the squares of the perpendicular distances from the sample points to the line is a minimum [Webb, 2003]. PCA is used on datasets when:

- The number of features needs to be reduced but removeable features can't be identified
- Features are ensured to be independent from on another
- It is suitable to make the features less interpretable [Murphy, 2012]

Yaslan and Cataltepe [2006] introduced PCA into music genre classification. PCA was applied to reduce the number of features in a set similar to one introduced by Tzanetakis and Cook [2002]. This consisted of 30 features with 6 beat, 9 STFT, 10 MFCC and 5 pitch features. A k-nearest-neighbour algorithm was used tested on the GTZAN dataset. Instead of cross validation, the data was split into a 90% training set and 10% test set with the training and testing procedures being repeated 30 times to obtain an average. Results found a classification accuracy of 70% with a single principle component. The accuracy increased to 80% when 15 principle components were used and stayed about this accuracy as the number of components was increased to 30.

Baniya and Lee [2016] looked into comparing PCA with the minimum redundancy maximum relevancy (MRMR) feature selection algorithm. This algorithm assigns a score to each feature based on its maximum relevance score (how relevant the feature is at describing some data). The maximum relevance score is calculated by taking the difference between the mutual information of the feature to the output (relevance), and the average redundancy of the selected feature to the whole feature set [Fuhui et al., 2005].

PCA and MRMR were compared using the GTZAN dataset using 118 extracted features consisting of dynamic, rhythmic and timbral features. The features were analysed over multiple frames to extract high level statistical features including the skewness and kurtosis. An SVM classifier was used with the results obtained using cross validation. A base classification accuracy of 80.75% was achieved using all the features. Results found that MRMR was able to provide a greater classification accuracy with more features removed compared to PCA. MRMR produced an accuracy of 87.9% after removing 37.2% of features while PCA produced an accuracy of 78.3% after removing 32.5% of features.

PCA offers an effective dimensionality reduction, however it lacks the complexity to asses if features provide a useful descriptor of genre. Due to the unsupervised nature of the algorithm, the first principle component may not always provide the strongest relationship of genre.

4.1.2. ReliefF

ReliefF is filter based algorithm feature selection algorithm based on Relief but applied to multiclass problems. ReliefF and Relief orders features according to their importance or *feature quality*. These feature statistics are referred to as feature weights ($W[A] = weight\ of\ feature\ A$). Feature sets are evaluated by searching for k of its nearest neighbours from the same class (called nearest hits) and k of its nearest neighbours for each different class (called nearest misses). $W[A]$ is then updated for all for all attributes A depending on their values for the nearest hits and nearest misses. ReliefF suffers from

two main limitations. It will assign higher weights to all features which have a high correlation to classification, so it can't effectively remove redundant features. It's also possible it will remove some features which have low weights but will get better classification results combined with other features.

Wu and Wang, [2015] proposed a feature selection algorithm based on the ReliefF and SFS algorithms for use in music genre classification. Sequential Forward Search (SFS) is a heuristic search algorithm. It adds features incrementally to a model if it improves the classification accuracy. The algorithm stops when the accuracy cannot be improved by the remaining features, or there is a limit to the maximum number of features it can use. SFS suffers from low performance (most notable on large feature sets) and once a feature is added to the model, it cannot be removed. ReliefF-SFS aims to improve on these drawbacks. First, the ReliefS-SFS algorithm aims to calculate the weights of all the features by ReliefF. It then tries to add the feature into the optimal feature set from highest weight to lowest. If the added feature improves the classification accuracy, it is added to the feature set. If it doesn't the feature is removed. This system can be described as a hybrid of wrapper and filter methods, ReliefF being a filter method and SFS a wrapper method. This was implemented with an SVM classifier using a feature set consist of the mean and variance of 12 MFCC and MFCC difference coefficients. The dataset used is not specified but describes a GTZAN like set consisting of five genres. A classification accuracy of 91.2% was reported using the ReliefF-SFS algorithm compared with 84.8% using no feature selection.

4.2. Metaphor-based Metaheuristic Feature Selection Algorithms

Metaphor-based metaheuristics represent a set of algorithms inspired by nature or the surrounding world to provide a sufficiently good solution to an optimisation problem. Firstly, existing use of metaheuristics in music genre classification will be presented. Two further algorithms will be presented that have seen use in feature selection but not been applied to music genre classification.

4.2.1. Self-Adaptive Harmony Search Algorithm

Huang et al. [2014] proposes a genre classification system based on feature selection using the Self-Adaptive Harmony Search algorithm (SAHS). SAHS is based on a metaheuristic algorithm called Harmony Search (HS) which is used for optimising search algorithms.

Harmony Search was introduced by Geem et al. [2001] and is based on the improvisation process of jazz musicians. Harmony search uses the following metaphors:

- Harmony represents a feature subset
- Harmony Memory is collection of feature subsets
- Notes represent features.

A group of solutions is randomly generated for an arbitrary problem and stored in harmony memory (HM). A new solution is generated based on those in the HM. If this solution is better than the worst

solution in the HM, the worst solution is replaced by this new solution. This process is continued until the maximum number of iterations has been reached. The system can be altered using the parameters of the pitch adjustment rate (PAR) and variable distance bandwidth (bw). PAR determines whether further adjustment is required according to bw.

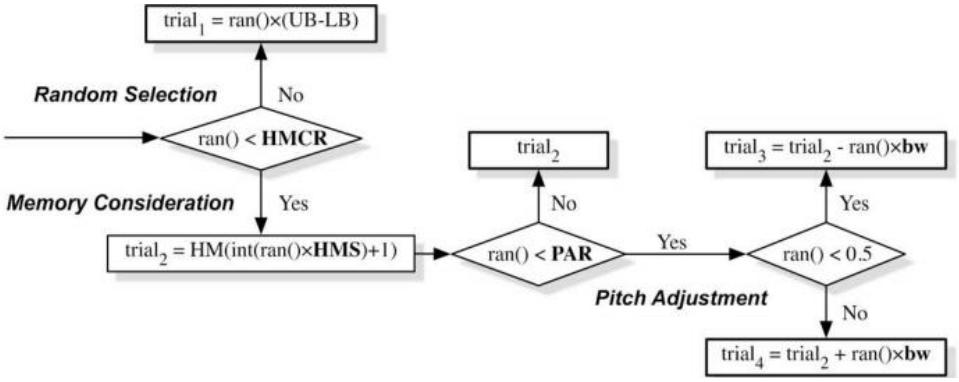


Fig 10: Block diagram of the Harmony Search algorithm. Originally from [Huang et al., 2014].

SAHS alters the pitch adjustment section of the algorithm to target an optimal solution. The bw parameter is replaced by updating the new harmony according to the minimal and maximal values within the HM. Finer adjustments are made which allows for the harmony to reach an optimum gradually. [Huang et al., 2014] tested the SAHS algorithm using a feature set consisting of intensity, pitch, timbre, tonality and rhythmic features. This resulted in a feature set of over 265 features. An SVM was implemented using an OVO approach. SAHS was implemented for each binary classifier to produce a feature set for each pairwise set of genres. The aim was to find features that better define specific genre pairs e.g. does tempo or key differ more in rock and reggae music? Experiments using the GTZAN dataset were conducted. The SAHS feature selection algorithm with a wrapper approach produced an accuracy of 97.2%.

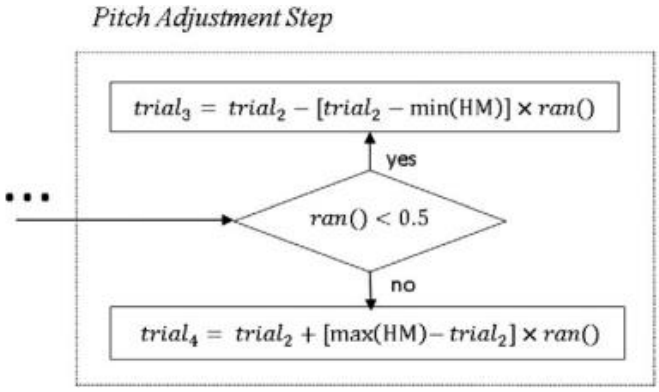


Fig 11: Pitch adjustment step of the SAHS algorithm. Originally from [Huang et al., 2014].

The performance of this system is significantly improved compared to traditional systems. This is likely due to the individualised feature sets for binary classifiers combined with the SAHS feature selection

algorithm. Further studies have ignored metaheuristic feature selection algorithms in combination a with one versus one approach, opting for research into the use of neural networks for classification which have failed to outperform this method [Fulzele et al., 2018; Rajanna et al., 2015].

4.2.2. Cuckoo Search

Cuckoo search (CS) is another metaheuristic optimisation algorithm developed by Yang and Suash Deb [2009]. It is inspired by species of cuckoos that lay their eggs in the nests of other species of birds. Cuckoo search uses the following metaphors:

- Each egg in a nest represents a solution.
- A cuckoo egg represents a new solution.

The aim is to use the new and potentially better solutions to replace an inferior solution in the nest.

Cuckoo search is based on three rules:

- Each cuckoo chooses a random nest to lay eggs
- The number of available nests is fixed. Nests with the highest quality eggs will carry on onto the next generation/iteration
- In the case a host bird discovers the cuckoo egg, it can discard the egg or abandon nest completely and build a new one.

An initial population for n host nests is first generated. For each iteration of the algorithm, a new solution is generated using a process known as Lévy flights. This is a random walk in which the step-lengths have probability distribution that is heavy-tailed. This results in lots of small steps followed by a few large steps (Fig 12). This allows the CS algorithm efficiently to explore the search space as its step length is much longer in the long run [Rodrigues et al., 2013]. The nests which have eggs of the lowest quality are replaced according to a probability of $p_a \in [0,1]$.

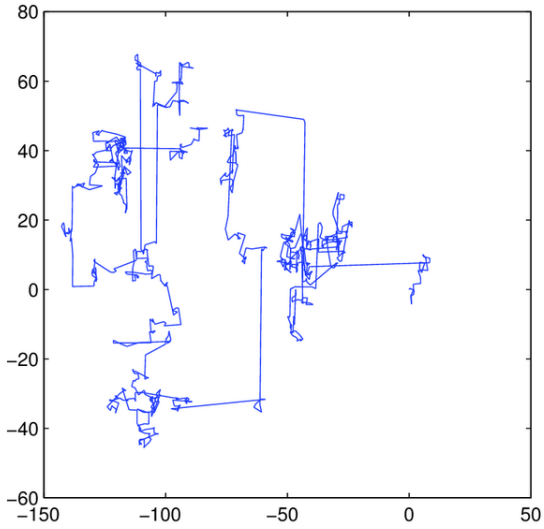


Fig 12: Example of a Lévy flight in two dimensions. Originally from [Pang et al., 2018].

Traditional CS updates the solutions towards continuous-valued positions. Binary Cuckoo Search (BCS) is where the algorithm is searching for an optimal binary code n bits long. BCS can be used in feature selection by using the binary code to represent what features to be used. The binary code represents whether to include a feature or not (1 is include a feature, 0 is ignore a feature) and as such n will be equal to the total number of features. Once the maximum number of iterations or stopping criteria has been reached the algorithm will return the highest performing feature subset. Using BCS for feature selection was originally proposed by Rodrigues et al. [2013] and found it to perform similarly to other metaheuristic feature selection algorithms but outperform harmony search. As of writing it has not been used in music genre classification but has seen successful use in image classification and theft detection in power distribution systems [Medjahed et al., 2015; Rodrigues et al., 2013].

4.2.3. Dragonfly Algorithm

The Dragonfly Algorithm (DA) is another metaheuristic algorithm developed by Mirjalili [2016] inspired by the swarming behaviour of dragonflies. It uses the following metaphors to replicate this behaviour:

- Separation, which refers to the mechanism dragonflies employ to avoid collisions with other dragonflies in the neighbourhood
- Alignment, which indicates velocity matching of dragonflies to that of other dragonflies in the neighbourhood
- Cohesion, which refers to the tendency of dragonflies movement towards the centre of mass of the neighbourhood

Attraction towards the food source and escaping from enemies are two other key behaviours that dragonflies exhibit to survive. The behaviour of dragonflies is assumed to be a combination of these five patterns.

DA utilises a step vector and position vector to solve optimisation problems. The step vector shows the direction of movement of the dragonflies and is defined as follows:

$$\boldsymbol{\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t} \quad \textbf{(10)}$$

where s shows the separation weight, S_i indicates the separation of the ith individual, a is the alignment weight, A_i is the alignment of ith individual, c indicates the cohesion weight, C_i is the cohesion of the ith individual, f is the food factor, F_i is the food source of the ith individual, e is the enemy factor, E_i is the position of enemy of the ith individual, w is the inertia weight, and t is the iteration counter. The position vectors are calculated as follows:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (11)$$

where t is the current iteration.

A set of random solution is initially generated. The position and step vectors of dragonflies are also initialised by random variables defined within the lower and upper bounds of the variables. In each iteration, the position and step of each dragonfly is update using equations (10) and (11). The position updating process is continued iteratively until the maximum number of iterations has been reached or until the end criterion is satisfied.

Similar to Cuckoo Search, the Dragonfly Algorithm updates solutions towards continuous valued positions. The Binary Dragonfly Algorithm (BDFA) has been adapted to use the Dragonfly algorithm for feature selection. Mafarja et al. [2017] proposed the BDFA algorithm for feature selection using a binary code to represent the feature subset to use in a system. This was evaluated on 18 different datasets and compared with the two other metaphor metaheuristics, Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO). The results showed the BDFA outperformed the GA and PSO feature selection algorithms. BDFA has yet to be implemented in music genre classification.

4.3. Summary

Features selection aims to solve the problem of providing a classification algorithm with strong genre correlated features. Traditional methods of feature selection along with metaphor-based metaheuristic methods have been presented.

What traditional feature selection algorithms have been used in music genre classification?

Principal Component Analysis and ReliefF are feature selection algorithms that have seen use in music genre classification. Research has shown improved performance of both PCA and ReliefF-SFS over an original feature set but neither have outperformed Huang et al. [2014] use of SAHS.

What metaheuristic feature selection algorithms have been used in music genre classification?

Harmony Search is the only metaheuristic that has seen use in music genre classification. Its implementation saw significant improvements in classification accuracy. However, it is unclear if its success is specifically due to the implementation of the SAHS algorithm or due the individual feature sets for binary classifiers. Further research is needed to confirm the use of metaheuristic algorithms within music genre classification.

What further metaheuristic feature selection algorithms can be used in music genre classification?

Two metaphor-based metaheuristic feature selection algorithms have been presented. Binary Cuckoo Search uses the behaviour of cuckoos laying their eggs in the nest of other birds' species to find optimal features in a feature set. The Binary Dragonfly Algorithm is a recently introduced metaheuristic that is based on the swarming behaviour of dragonflies. Neither BCS nor BDFA have been implemented as feature selection algorithms in music genre classification but have seen successful use in feature selection for other classification tasks.

Chapter 5
Hypothesis

5.1. Literature Review Conclusion

The previous chapters have discussed the use of machine learning in music genre classification. Machine learning has been applied to music genre classification to automatically predict the genre of an unknown piece of music. This is achieved by extracting features from a large number of samples to train a classification model. Features shown to have a correlation in genre have been split into different timbral, rhythmic and tonal features. MFCC's and SSD's have been used as timbral descriptors. Chromagrams and Tonal Centroids have been used as tonal descriptors. Rhythm patterns show improved results over rhythm histograms and pattern clusters as rhythmic features. Support Vector Machines have shown good performance in classification of genre and continued to be used in state-of-the-art systems. All classification algorithms rely on strong genre correlated features for accurate predictions.

Chapter 4 looked at feature selection in music genre classification and the use of metaheuristic algorithms. These algorithms sort through the features to provide the most strongly correlated features. This can lead to more robust models, improved classification accuracy and a reduced time to train. PCA and ReliefF-SFS have been highlighted as feature selection algorithms that have seen use in music genre classification. SAHS is metaphor-based metaheuristic that has been used as a feature selection algorithm in music genre classification and shown significantly increased results. This particular implementation used feature selection to find feature sets for pairwise combinations of genres. Further metaphor-based metaheuristics BCS and BDFA have been highlighted along with their use in feature selection but neither applied to music genre classification.

Research into use of metaheuristic feature selection algorithms, both in terms of implementation and different algorithms could provide further improved results and confirmation of previous literature. Notable increases in performance have also been seen when finding an individual feature subset for each binary classifier. Again, this method has seen less coverage in the literature so further testing should validate the method.

Following the literature review, there are two research question that arise from this review thus far:

Do metaphor-based metaheuristic feature selection algorithms outperform traditional feature selection algorithms?

Little literature exists based on using metaphor-based metaheuristics for feature selection in music genre classification. What literature does exist though, reports that large improvements of performance have been found. Other classification problems have seen improved results using metaphor-based metaheuristics. Investigating whether these algorithms do improve the performance will provide further reason for their use in other classification problems. It is expected that these algorithms will outperform the traditional feature selection algorithms, due to them being able to explore a greater search space of possible feature subsets. SAHS is the only metaheuristic applied to music genre classification but has seen substantially improved results. Research using BCS and BDFA has found improved performance on other classification problems compared to using HS.

Does providing an individual feature set for each binary classifier improve the performance of the system?

Current research has focused on providing a single subset of features for a classification system. While this has shown improved results, individual feature sets have produced a further increase in performance. This may be due to individual feature sets building more robust binary classifiers which may increase the performance of the system overall.

5.2. Hypothesis

By examining the performance of metaphor-based metaheuristic feature selection algorithms, hypotheses can be formed from the research questions:

1. Metaphor-based metaheuristic feature selection algorithms will outperform traditional feature selection algorithms and systems using all available features.
2. Creating individual feature sets for binary classifiers using metaphor-based metaheuristics will outperform a system using all available features or a selected subset.
3. In both implementations, Binary Cuckoo Search and Binary Dragonfly algorithm will produce a feature set that outperforms a set found using Self Adaptive Harmony Search.

Chapter 6
Experiment

In order to test the hypotheses outlined in Chapter 5, an experiment assessing the effectiveness of meta-heuristic feature selection algorithms was carried out.

Two experiments were conducted; the first experiment compared the use of traditional and metaphor-based metaheuristic feature selection algorithms. The FS algorithms used were PCA, ReliefF-SFS, SAHS, BCS and BDFA. The second experiment compared just the metaphor-based metaheuristic feature selection algorithms using an OVO approach but with individual feature sets being selected for binary classifiers. The FS algorithms used were SAHS, BCS and BDFA. Feature sets using the FS algorithms were extracted before the testing procedure, so the measured test time doesn't include the time taken by the FS algorithms to run and search for the optimal feature sets.

To test feature selection algorithms, annotated datasets are required, consisting of genre labelled music. From this, a labelled feature set can be extracted. A feature set consisting of timbral, rhythmic and tonal features was extracted from two datasets:

1. **GTZAN** [http://marsyas.info/downloads/datasets.html, 2001; Tzanetakis and Cook, 2002]: This consists of 1,000 thirty second clips of songs split across equally over ten different genres: rock, pop, blues, country, disco, hip hop, metal, classical, jazz and reggae. All clips are mono AU format, 22.05kHz, 16 bit.
2. **ISMIR2004 [**http://ismir2004.ismir.net/genre_contest/, 2004]: This consists of 1,426 songs over six different genres: classical (637 songs), electronic (201 songs), jazz_blues (52 songs), metal_punk (90), rock_pop (201 songs) and world (244 songs). All clips are stereo MP3 format, 44.1kHz.

Feature extraction was achieved using the *librosa* Python library [https://librosa.github.io/librosa/index.html; McFee et al., 2015] and *MIRToolbox* MATLAB toolbox [https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox; Lartillot et al., 2008]. The final feature set consisted of 428 individual features consisting of timbral, tonal and rhythmic features. This consisted of MMFC's, SSDs, Rhythm Patterns and Chromagrams. The full list of features can be found in the appendix. All features were scaled between -1 and 1. All features were extracted

using a sampling rate of 22.05kHz, taking a 30 second mono sample around the middle of the audio sample.

The metaheuristic algorithms Self Adaptive Harmony Search (SAHS), Binary Cuckoo Search (BCS) and Binary Dragonfly Algorithm (BDFA) were implemented for feature selection. For all metaphor-based metaheuristics, the number of iterations was set to 300 and the initial solution population to 20. The HMCR of the SAHS algorithm was set to 0.99. These values were chosen to be similar to those of the studies in the literature [Huang et al., 2014; Mafarja et al., 2017; Rodrigues et al., 2013]. Traditional feature selection algorithms Principal Component Analysis and ReliefF-SFS were also implemented to compare the performance of metaheuristics against traditional feature selection algorithms. The PCA variance hyperparameter was set to 0.98. The number of neighbours for ReliefF-SFS was set to 100.

An SVM classifier was implemented using an OVO approach. Grid search was used to find the optimal hyperparameters. This resulted in a penalty error C of 3 and a gamma of 0.02 using an RBF kernel. The models were programmed in Python using *scikit-learn* [Pedregosa et al., 2011] for the machine learning algorithms and evaluation. Statistical analysis was conducted using *scipy* [Jones et al., 2001]. Results were plotted using the *matplotlib* Python library [Hunter, 2007]. Full code and can be found on the included DVD's or at https://github.com/joshprewer/MusicGenreClassification. The models were trained and tested ten times using k-fold cross validation using ten splits for the GTZAN datasets and six splits for ISMIR2004 dataset.

All processing, training and testing was done on a Macbook Pro (2015), with an i7 processor and 16GB RAM, running macOS Mojave.

Chapter 7
Results and Discussion

This experiment assessed the following hypotheses:

1. Metaphor-based metaheuristic feature selection algorithms will outperform traditional feature selection algorithms and systems using all available features.
2. Creating individual feature sets for binary classifiers using metaphor-based metaheuristics will outperform a system using all available features or a selected subset.
3. In both implementations, Binary Cuckoo Search and Binary Dragonfly algorithm will produce a feature set that outperforms a set found using Self Adaptive Harmony Search

This chapter discuss the results of the experiment and is split into three sections. First, a comparison of the traditional and metaphor-based metaheuristics is given. Following this, analysis is made regarding the use of the Self-Adaptive Harmony Search, Binary Cuckoo Search and Binary Dragonfly Algorithm for finding individual feature sets for binary classifiers. Lastly, retrospective discussion about the results and experiment procedures is made and possible improvements suggested.

7.1. Traditional and Metaphor-Based Metaheuristic Algorithms

The first round of testing looked at the performance of the five different feature selection algorithms on both the GTZAN and ISMIR dataset. To effectively compare the results, a normality test was done to show if the data is normally distributed. This allows an ANOVA test and other parametric tests to be computed to compare the significance of the different results. This was achieved by using the normality test function in *scipy* which is based on [D'Agostino and Pearson, 1973]. These results are shown in Table 1. A p-value of less than 0.05 indicates the results may not be normally distributed. This was the case with the PCA and BDFA ISMIR results. The remaining results show a normal distribution making parametric analysis suitable. The performance of these algorithms is seen in Table 2, where the highest F1 score and lowest test time is emboldened. The F1 score was used as the performance metric to take into account the balanced GTZAN dataset and unbalanced ISMIR dataset. The values represent the mean of the results. For the sets of results that failed to show a normal distribution, the median of the performance is included in brackets.

	GTZAN	**ISMIR**
No FS	0.886	0.097
PCA	0.778	**0.038**
ReliefF-SFS	0.781	0.219
SAHS	0.365	0.322
BCS	0.607	**0.041**
BDFA	0.977	**< 0.001**

Table 1: P-values of normality test for the different F1 score results. Results under 0.05 are emboldened.

	GTZAN		**ISMIR2004**	
	F1 Score	Time (ms)	F1 Score	Time (ms)
No FS	0.819	642	**0.782**	632
PCA	0.816	223	0.77 (0.776)	246
ReliefF-SFS	0.762	**104**	0.58	**90**
SAHS	0.812	250	0.754	292
BCS	**0.823**	343	0.765 (0.77)	399
BDFA	0.818	497	0.773 (0.781)	422
p-value	< 0.001		< 0.001	
p-value*	0.384		0.012	
p-value**	0.16		0.015	

Table 2: The mean F1 scores and test times of five different feature selection algorithms. * this p-value compares all results asides from ReliefF-SFS. ** this p-value compares SAHS, BCS and BDFA.

Looking at the results of Table 2, the differences between the feature sets is small. However, a large decrease in performance is seen in the ReliefF-SFS feature set, which scored the lowest F1 score but had the quickest test time. As shown by Fig 15, this performance is likely due to the ReliefF-SFS feature set having the lowest number of features. This large decrease in features may be due to the algorithm failing to factor in how features can be used together to classify genre as opposed to individually.

P-values were calculated to provide an answer to the first and third hypotheses. The results of ReliefF-SFS heavily skewed the p-value. Further p-values were computed, one comparing all the results asides from ReliefF-SFS (p-value*) and another comparing just the SAHS, BCS and BDFA results (p-value**). The p-values for the time results weren't included as the focus on the experiment was classification performance.

Aside from Relief-SFS, the algorithms performed differently using the two genre datasets. Looking at the GTZAN results, BCS and BDFA feature sets outperformed the other feature subsets with improved

F1 score and decreased test time. However, only BCS outperformed the performance of the full feature set with a F1 score rise of 0.004 and being 299ms quicker in making predictions. The SAHS and BDFA feature set failed to outperform the base system. However, BDFA underperformed the base system by a drop in F1 score of 0.001. Similar to the ReliefF-SFS feature set, this lack of performance may be due SAHS selecting the fewest features out of the metaphor-based algorithms. The second p-value of the results show that difference between the metaheuristics feature selection algorithms and traditional algorithms is statistically insignificant to prove the first hypothesis. The third p-value looks at the differences of just the metaheuristic algorithms, which shows there are stronger results to suggest that BCS and BDFA outperform SAHS but fails to reach the statistically significant mark of 0.05.

The ISMIR dataset produced contrasting results to those of the GTZAN dataset. BDFA provided the highest F1 score out of all the FS algorithms, but it failed to match the F1 score of the system using all available features. Nonetheless, all the algorithms showed an improved test time with little drop in F1 score. BDFA and BCS outperformed SAHS in both datasets but failed to match the performance of PCA. The difference in performance across datasets may be due to noise in the the datasets or the unbalanced nature of the ISMIR2004 dataset. The p-value of 0.012 shows that this decrease in performance is statistically significant, suggesting inverse of the first hypothesis. However, the third p-value of 0.015 suggests that the differences SAHS, BCS and BDFA is statistically significant to suggest that BDFA and BCS outperform SAHS.

Fig 13 and Fig 14 show the confidence intervals of the mean performance scores. All the test results of performance demonstrate a small confidence interval making the mean a good indicator of the scores of the data that is normally distributed. Ignoring the ReliefF-SFS results, Fig 13 shows the crossover of the confidence intervals which explains why the p-value for these results is so high.

Fig 15 and Fig 16 show the types of features that were selected from each algorithm. PCA features are their own separate type due to it combining features. The metaphor-based metaheuristic algorithms included the most features in their subsets. Across all feature subsets timbral features were the most included. It should be noted that the original feature set contained 244 timbral features, 77 tonal features and 107 rhythmic features. The results are similar across datasets aside ISMIR including no rhythmic features in any of the subsets. Of the GTZAN results, BDFA was the only algorithm to include rhythmic features in its feature set. From these results, rhythmic features appear to have the lowest impact in classifying genre compared to the other features available. In both of the datasets, BDFA produced the greatest number of features.

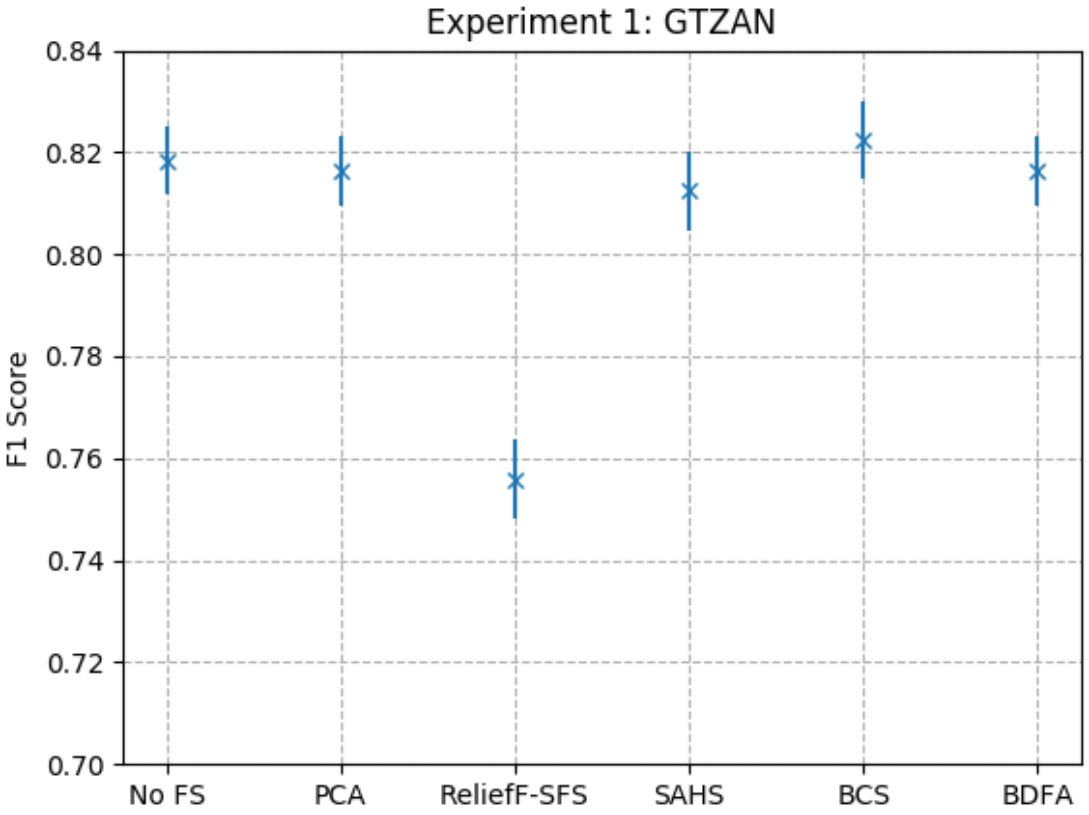


Fig 13: Mean performance and 95% confidence intervals of algorithms on GTZAN dataset

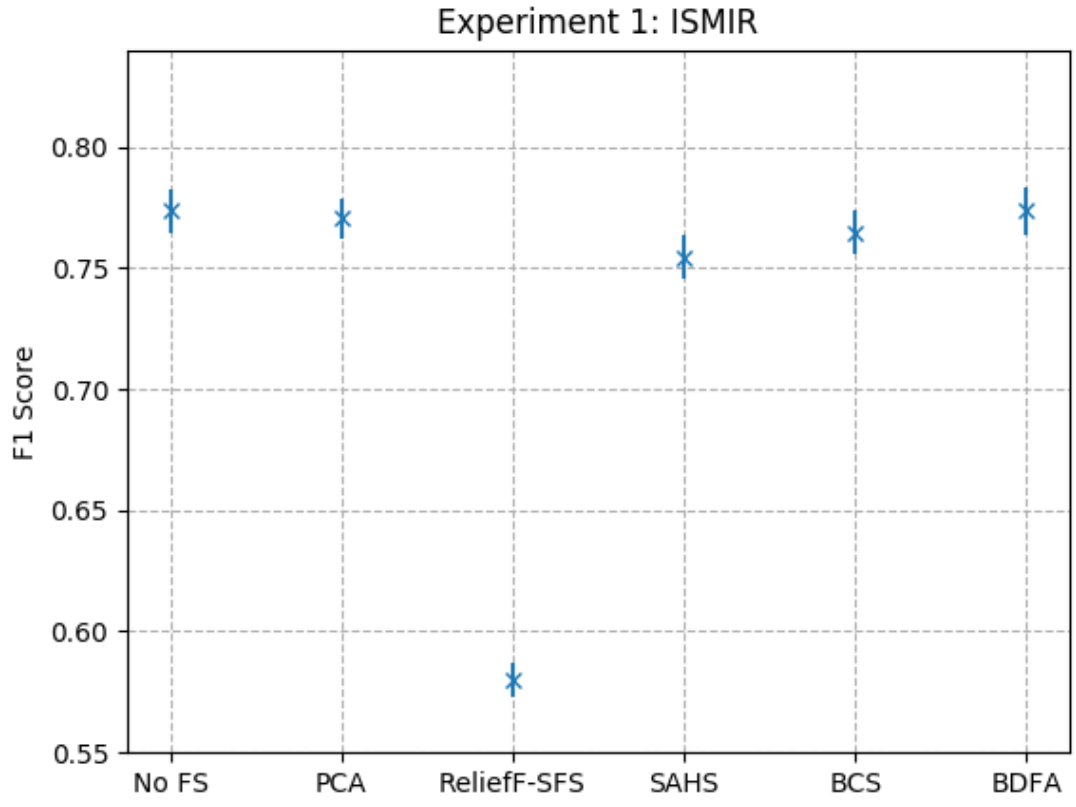


Fig 14: Mean performance and 95% confidence intervals of algorithms on ISMIR04 dataset

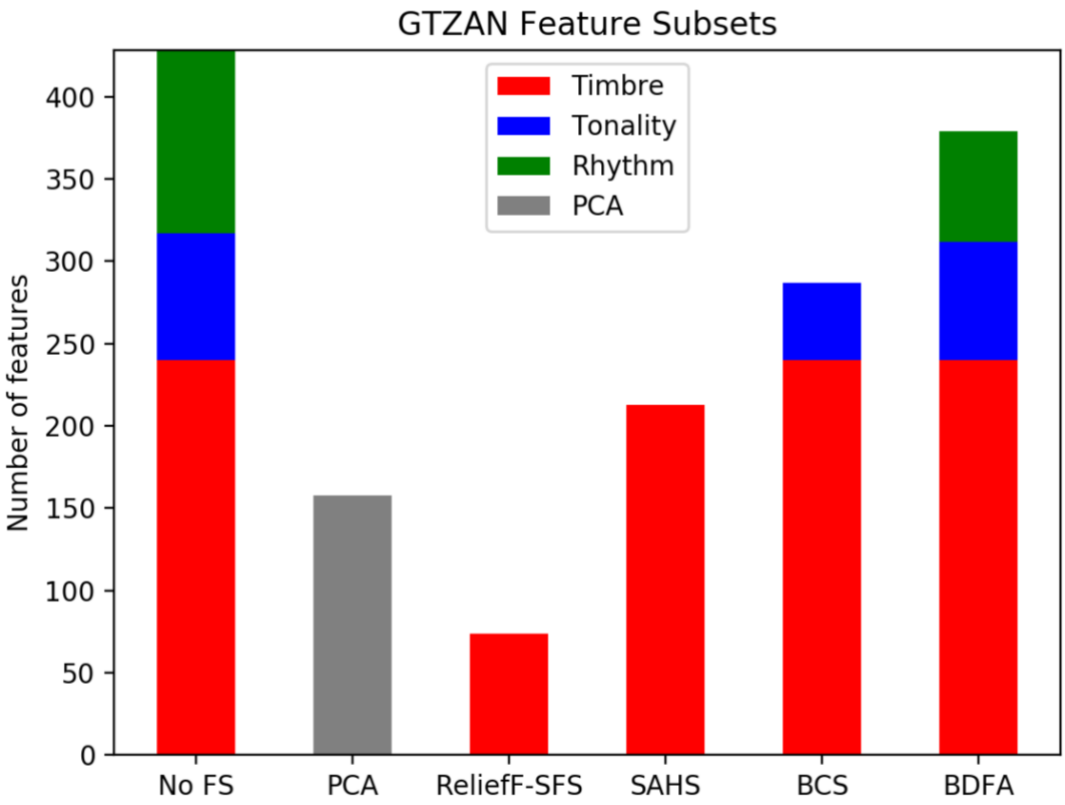


Fig 15: Table showing the features chosen by each FS algorithm

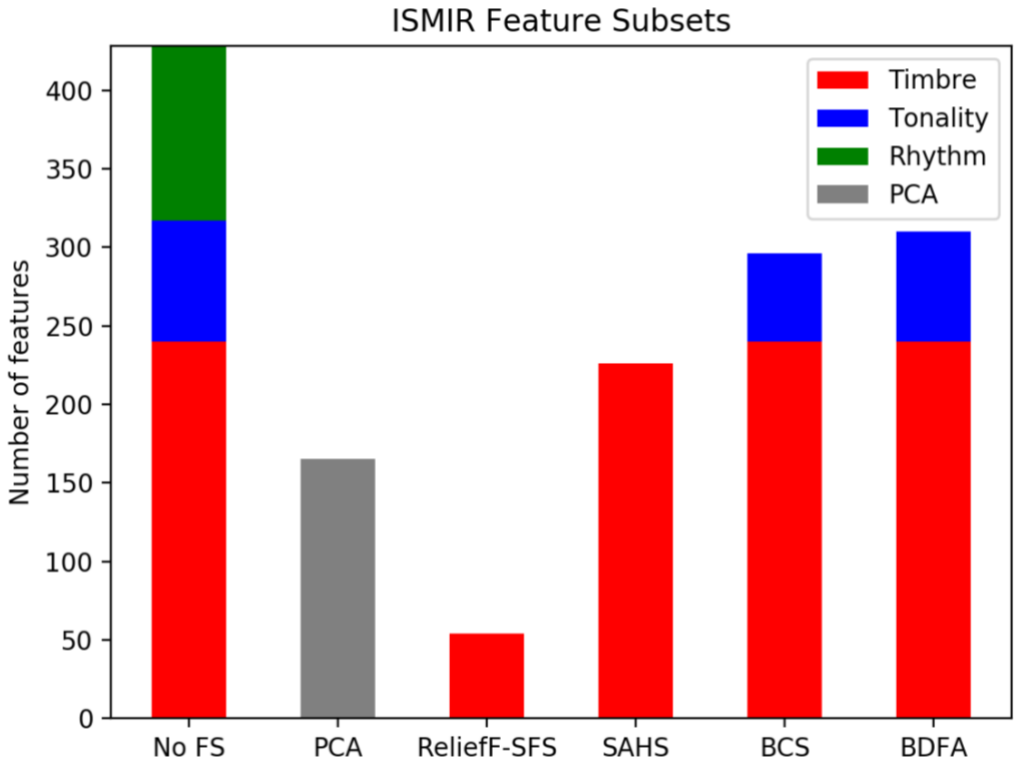


Fig 16: Chart showing the features chosen by each algorithm using the ISMIR dataset

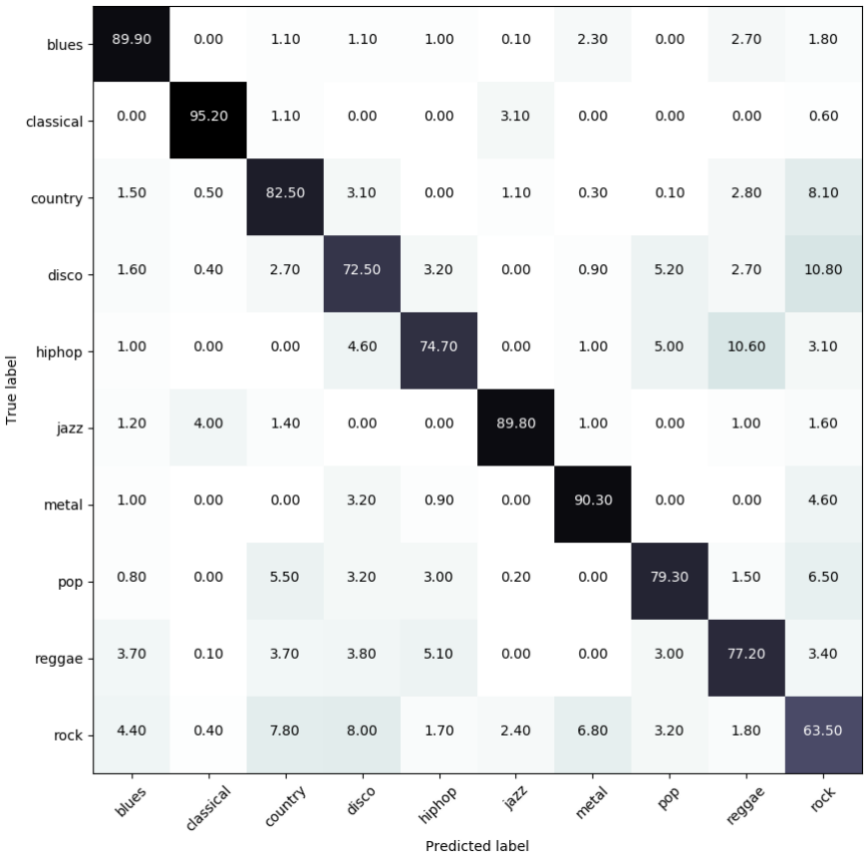


Fig 17: Confusion matrix of accuracy performance on GTZAN dataset using all features

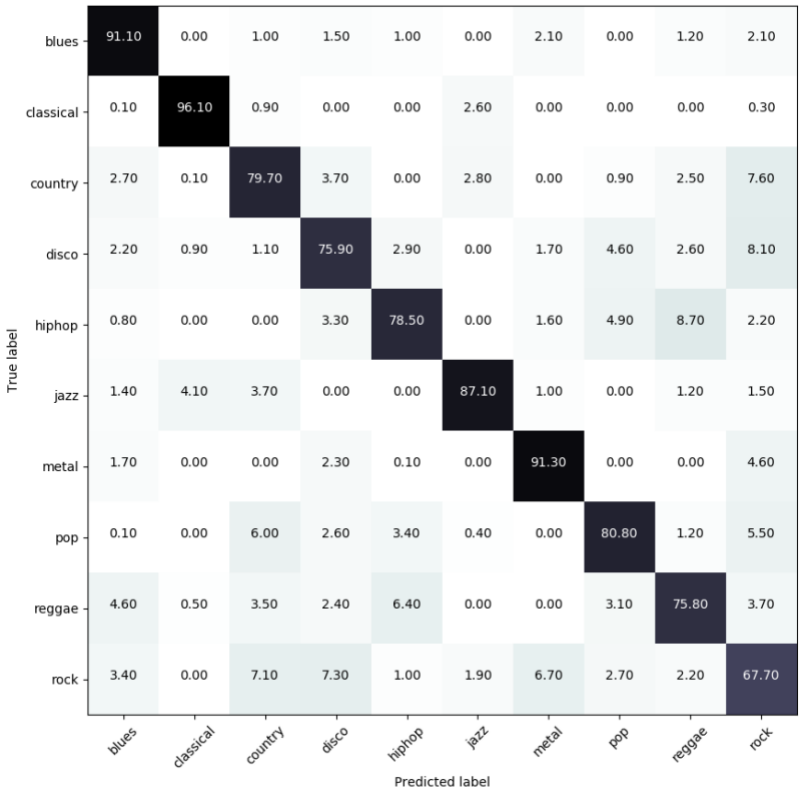


Fig 18: Confusion matrix of accuracy on GTZAN dataset using BCS feature set

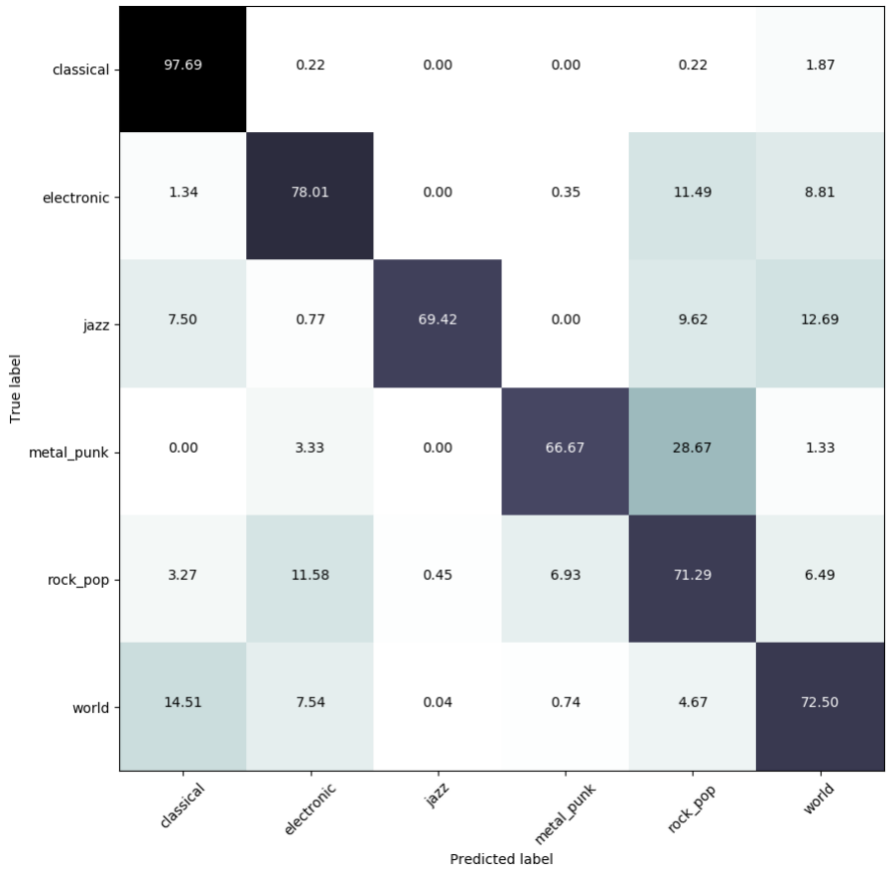


Fig 19: Confusion matrix of accuracy on ISMIR dataset using full feature set

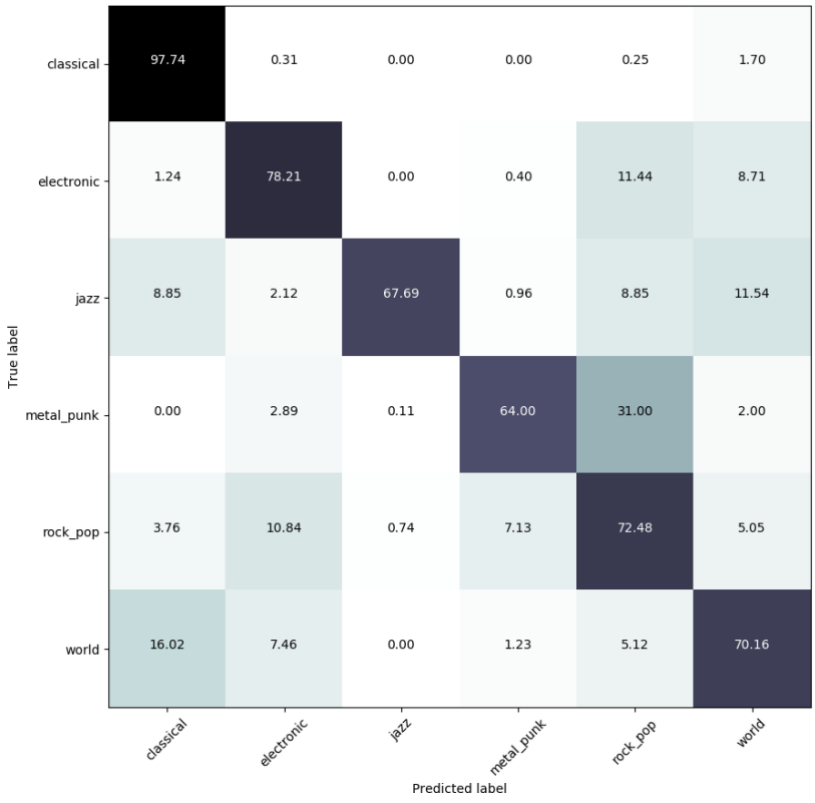


Fig 20: Confusion matrix of accuracy on ISMIR dataset using BDFA feature set

Further analysis can be done by looking at the confusion matrix of the algorithms. This highlights how each genre score in the system. These matrices do not have to be symmetrical as they plot the percentage of each predicted labels being a different class. As such, the sum of horizontal percentages for each true label will equal 100. Both the results shown have a weakness in classifying GTZAN rock, where it would be misclassified as country, disco or metal. The ISMIR dataset showed a similar strength in classifying classical compared to the GTZAN dataset. Furthermore, ISMIR had the most difficulty classifying metal_punk. Rock_pop wasn't the most ambiguous genre to classify but it would often mistake metal_punk for rock_pop. Looking at Fig 19 and Fig 20, BDFA feature set fails to significantly improve the classification accuracy of any genre. These genre weaknesses may potentially be alleviated by having individual features sets for binary classifiers.

Overall these results fail to demonstrate the hypothesised superior metaheuristic feature selection algorithms and prove the first hypothesis. Although a small increase in the GTZAN dataset was found, the ISMIR dataset failed to support this with the base feature set outperforming the other feature sets Furthermore, the GTZAN results were shown to be statistically insignificant compared to the ISMIR results, meaning the algorithms may not be responsible for the increased performance. The difference in performance across datasets may be due to a number of factors; either the quality of the actual data or the unbalanced/balanced nature of the datasets. However, the metaphor-based metaheuristic feature sets produced the highest performance out of all the feature sets across both data sets. Although there is insufficient evidence to prove the hypothesis, these feature selection algorithms have shown to perform better or match the performance of traditional feature selection algorithms. There is more evidence to prove the third hypothesis, with statistically significant results of BCS and BDFA outperforming SAHS on the ISMIR dataset. An increase was seen using BCS on the GTZAN but alongside the results of BDFA it wasn't statistically significant. Some genres have been shown to be more difficult to classify compared to others, notably rock in the GTZAN dataset and metal_punk in the ISMIR dataset. Experiment 2 aims to improve on these genre weaknesses.

7.2. Individual Feature Selection

This round tested the three metaphor-based metaheuristics using individual feature sets for each binary classifier. Table 3 shows the results of a normality test. Aside from the BDFA GTZAN results, all the data showed a normal distribution. These results are shown in Table 4. The highest F1 score and lowest test time are emboldened. P-values were calculated using all the results (p-value) and using just the SAHS, BCS and BDFA results (p-value*). The p-values for the time results weren't included as the focus on the experiment was classification performance.

	GTZAN	**ISMIR**
No FS	0.543	0.41
SAHS	0.999	0.99
BCS	0.418	0.4
BDFA	**0.011**	0.443

Table 3: P-values of normality test for the different F1 scores results. Results under 0.05 are emboldened.

	GTZAN		**ISMIR2004**	
	F1 Score	Time (ms)	F1 Score	Time (ms)
No FS	0.819	642	**0.782**	632
SAHS	0.822	**280**	0.734	**295**
BCS	**0.825**	368	0.754	371
BDFA	**0.829 (0.822)**	334	0.733	339
p-value	0.153		< 0.001	
p-value*	0.544		0.008	

Table 4: The mean F1 scores and testing time of metaheuristic algorithms on both datasets. * this p-value compares just SAHS, BCS and BDFA results

Again, these results differ with the genre dataset used. The GTZAN results demonstrate the hypothesised result. All three algorithms improved the classification accuracy compared with no feature selection. Furthermore, all the algorithms showed an improvement of performance compared with the results collected from the first experiment. SAHS saw the greatest increase with an F1 score rise of 0.01. However, these results do not demonstrate the greater hypothesised increase found in [Huang et al., 2014]. This lack of increase could be down to a number of difference factors. The implementation of Huang et al. [2014] method was attempted to be as similar to the paper, however some assumptions had to be made. This included the number of iterations of the feature selection algorithms and differences within the feature set. Out of the three algorithms tested, the BDFA had the greatest improvement to average F1 score with a 0.01 improvement over using the whole feature set and a 0.011

increase over using a single BDFA feature set for all binary classifiers. However, the median of the BDFA results puts it below the BCS result. The p-value for these results are more statistically significant than the GTZAN results of the first experiment but fail to meet the scientific measure of 0.05 to prove the second hypothesis. Again, the second p-value is statistically insignificant to prove the third hypothesis.

This improvement in results was not replicated using the ISMIR04 dataset. A drop in the F1 score occurred using each algorithm. BDFA had the most significant impact with a drop of 0.049 compared with using the full feature set and a drop of 0.04 compared to the single BDFA feature set from Table 2. Furthermore, none of the algorithms outperformed their counterpart results from Table 2. Again, this difference in results may be due to the quality of the dataset or its unbalanced nature. Nonetheless, all the algorithms provided a faster testing time of a comparable degree to the reduced time in experiment 1. Similar to Table 2, the first p-value for these results shows that they are statistically significant to prove the inverse of the second hypothesis. However, the second p-value shows that the results are statistically significant, suggesting that BCS will outperform BDFA and SAHS.

Fig 21 and Fig 22 show the confidence intervals of the mean performance scores. All the test results of performance demonstrate a small confidence interval making the mean a good indicator of the scores of the data that is normally distributed. The large cross over of confidence intervals in the GTZAN dataset shows that the range of results do not differ enough to be statistically significant.

Fig 23 and Fig 24 show the types of features that were selected from each algorithm. The average number of features was taken along with the 95% confidence interval. Again, the majority of features included were timbral followed by tonal and then rhythmic. SAHS also produced the smallest feature set which provided it with the fastest testing time. Again, these results highlight the lack of importance associated with rhythmic features in highlighting genre. Less of a difference is seen between the datasets. BCS this time was found as the algorithm to produce the largest features.

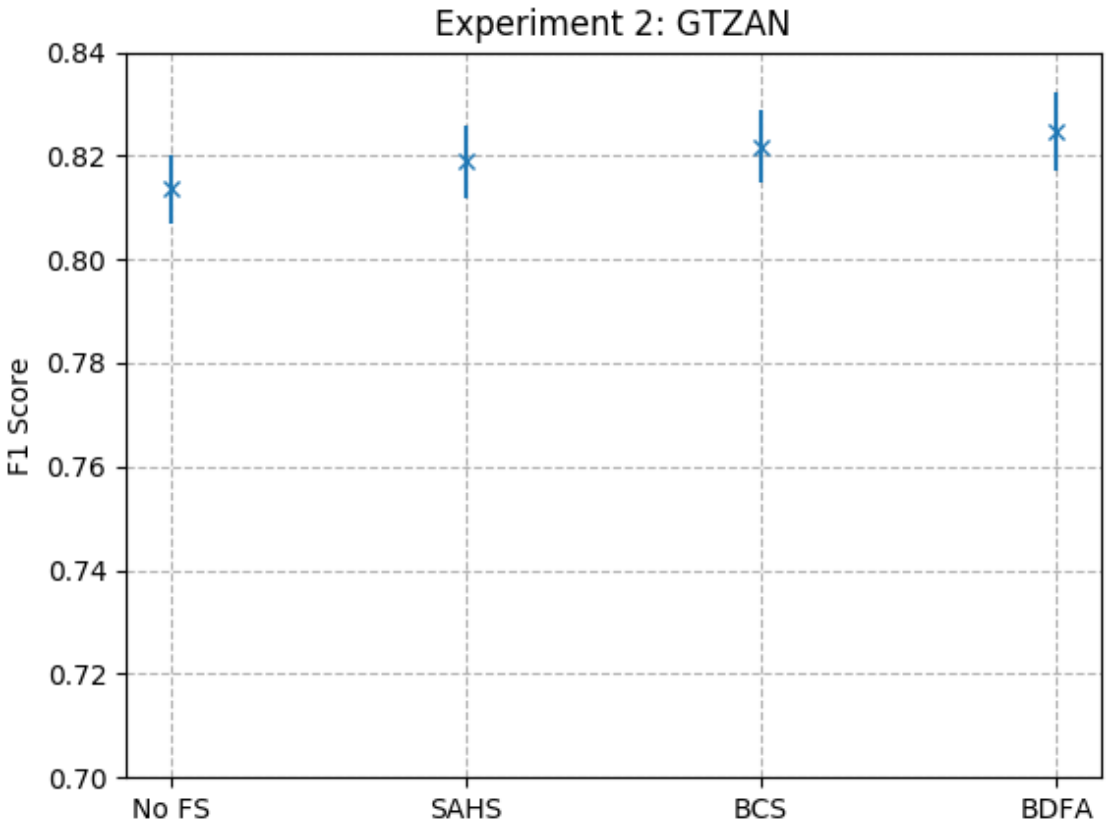


Fig 21: Average performance and 95% confidence intervals of algorithms on GTZAN dataset

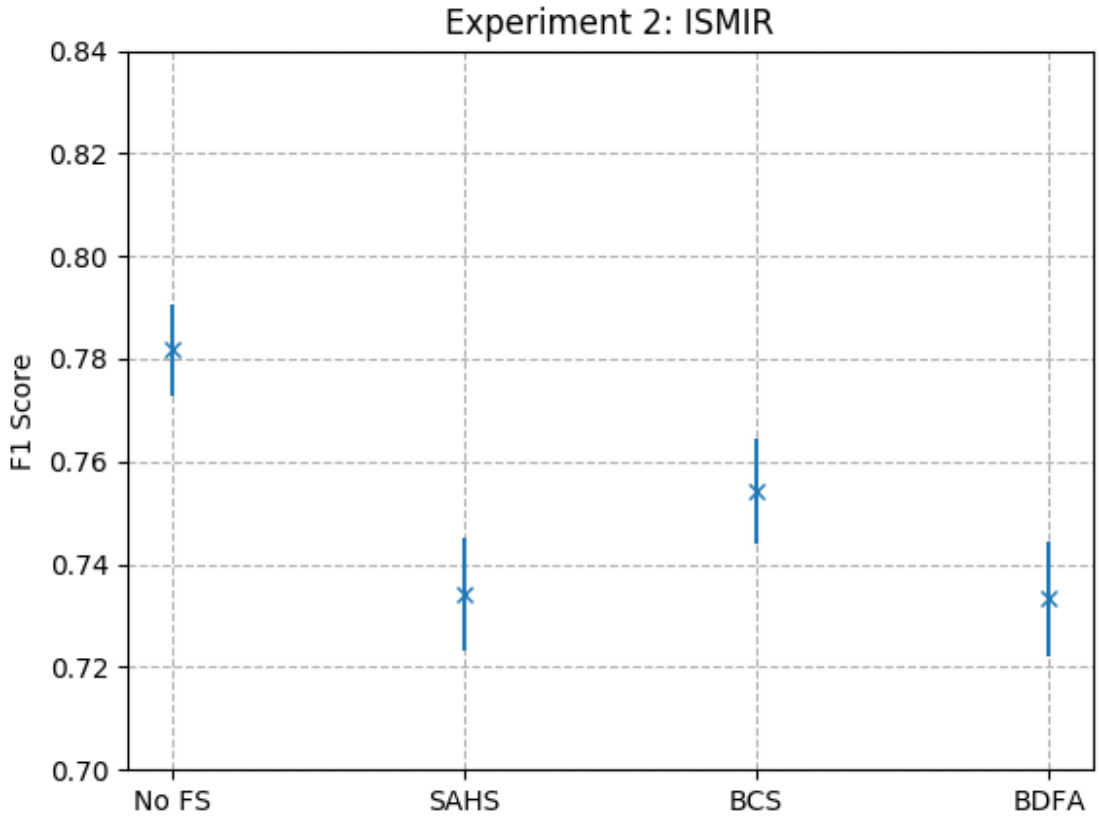


Fig 22: Average performance and 95% confidence intervals of algorithms on ISMIR04 dataset

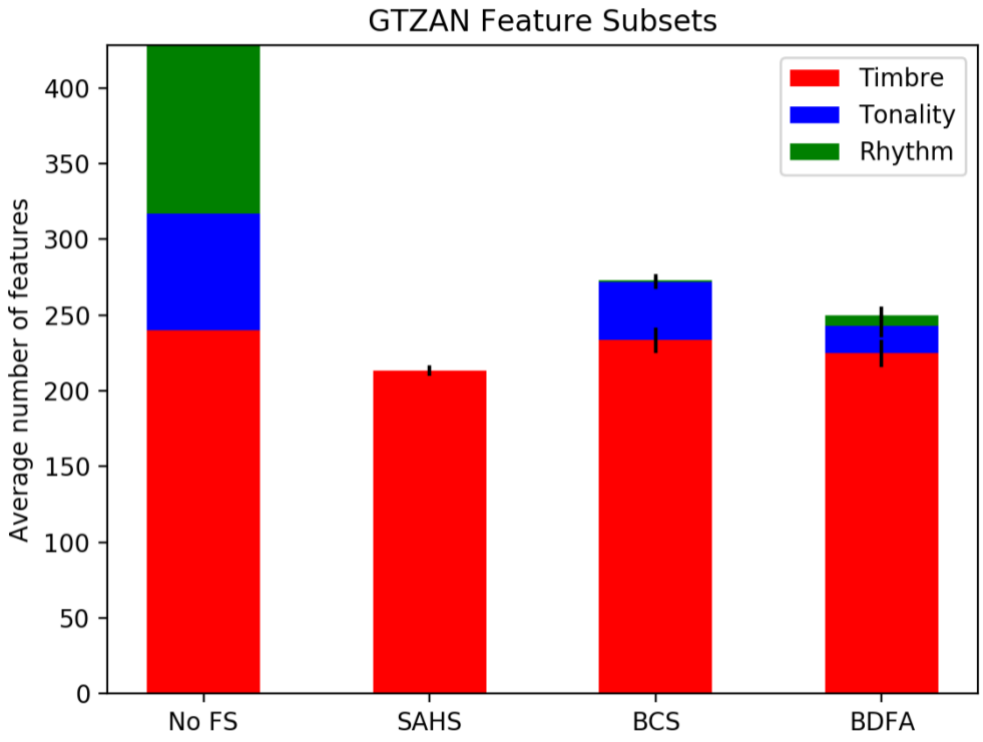


Fig 23: Chart showing the average number of features used in each binary classifier for the GTZAN dataset

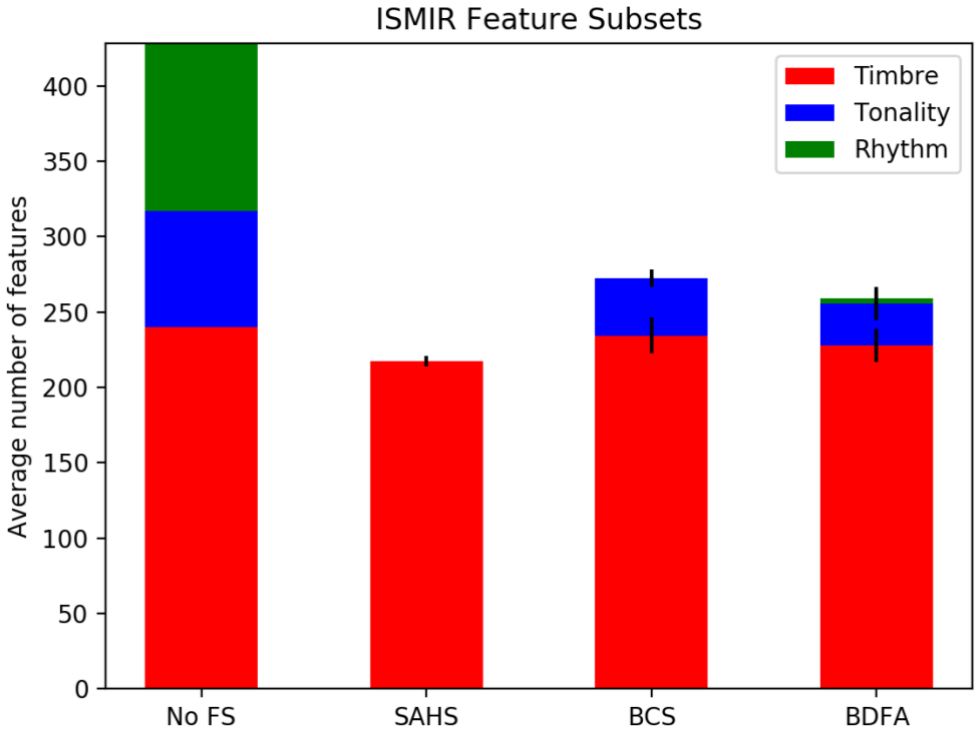


Fig 24: Chart showing the average number of features used in each binary classifier for the ISMIR04 dataset

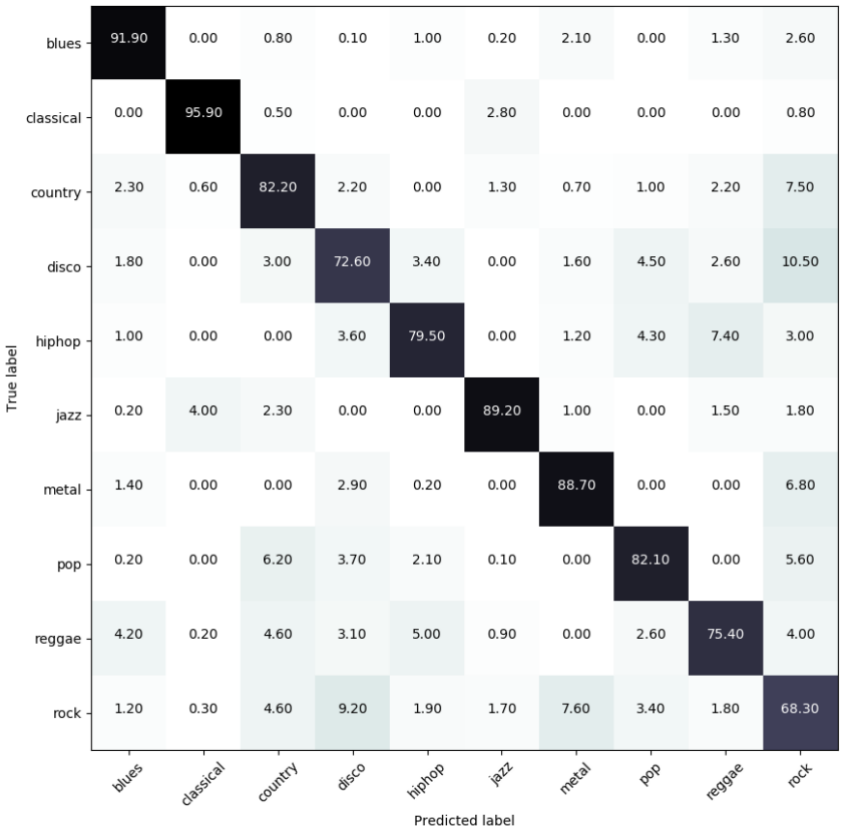


Fig 25: Confusion matrix of system performance with BDFA feature selection on binary classifiers

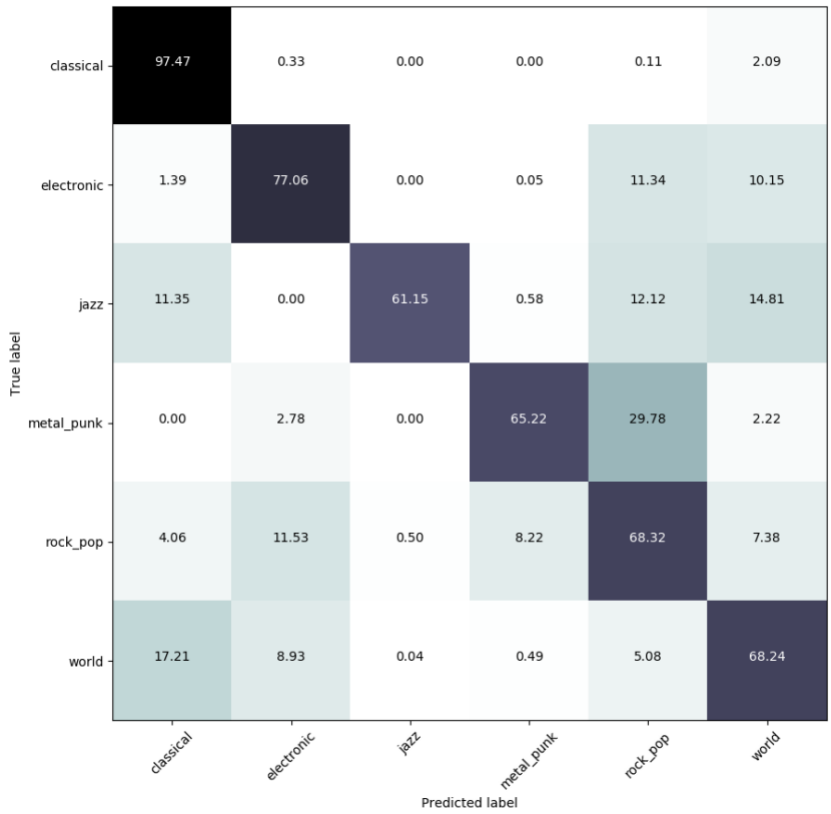


Fig 26: Confusion matrix of accuracy on ISMIR dataset using individual BCS feature sets

Further analysis of the algorithm's performance is seen in the confusion matrices of Fig 25 and Fig 26. Again, the GTZAN confusion matrix shows this implementation slightly increased the performance of classifying ambiguous genres rock and disco. As suggested by Table 4, this improvement in classifying ambiguous genres was not seen in the ISMIR confusion matrix aside from a 1.1% increase in classifying metal_punk.

Overall, the experiment produced mixed results, again dependent on the dataset used. The results of GTZAN dataset proved the hypothesis, with SAHS and BDFA outperforming the results from Experiment 1 with BCS only experiencing a 0.1% drop. Furthermore, BCS and BDFA outperformed SAHS. However, the ISMIR dataset failed to prove the second hypothesis with all algorithms dropping in performance compared to the first experiment. The ISMIR provided statistically significant results to suggest BCS outperformed SAHS and BDFA. However, combined with the insignificant GTZAN results it fails to prove the third hypothesis.

7.3. Summary

A number of conclusions can be drawn from these results. The experiments have shown that these different implementations and algorithms have a small influence over the results and are influenced by the genre dataset used. However, the statistical validity of these changes is questionable. In the cases where there was an improvement, it was less than hypothesised. Conversely, when the results contradict the hypothesis, the performance is reduced by a small margin. After recording the results, the author [Huang et al., 2014] was contacted for extra details on their SAHS implementation. Their experiment went through 10000 which is significantly higher than the 300 iterations used in the conducted experiment. This could explain the small improvement in results, as it would limit the algorithms from exploring further feature set combinations. Nonetheless, all results for the metaheuristic algorithms showed some applicability to features selection. All algorithms reduced the test time with a small impact on classification performance.

The three hypotheses presented in Chapter 5 can now be answered.

Metaphor-based metaheuristic feature selection algorithms will outperform traditional feature selection algorithms and systems using all available features.

The first experiment showed some evidence for a marginal improvement in results. However, it is unclear whether the performance of results is affected by the use of balanced/unbalanced datasets. Nonetheless, there is evidence to show that BCS and BDFA are effective as feature selection algorithms in music genre classification, by removing features and retaining a similar classification performance. It was noted that these algorithms tended towards including more features in the subsets. The distribution of these subsets relied heavily on timbral features. Further research in using these feature selection algorithms with a range of balanced and unbalanced datasets would be necessary to

demonstrate that these algorithms are as effective as traditional methods. The results fail to prove this hypothesis.

Creating individual feature sets for binary classifiers using metaphor-based metaheuristics can outperform a system using all available features or a selected subset

Again, the results fail to prove this. The algorithms showed an increase in performance on the GTZAN dataset with the BDFA providing the greatest increase compared with the other algorithms and single feature set implementation. However, this wasn't replicated on the ISMIR04 dataset. A drop in the F1 score occurred using each algorithm, with BDFA having the most significant impact. Again, further research in using individual feature sets using binary classifiers across balanced and unbalanced dataset would be necessary to provide sufficient evidence to prove this hypothesis.

In both implementations, Binary Cuckoo Search and Binary Dragonfly algorithm can produce a feature set that outperforms a set found using Self Adaptive Harmony Search

The ISMIR results provided the most statically significant set of results to prove this hypothesis. Experiment 1 showed an increase in the BCS and BDFA algorithm compared to SAHS with statistically significant results. However, the BCS and BDFA results failed to show a normal distribution so findings should be taken with consideration. Experiment 2 showed normally distributed results with BCS outperforming SAHS and BDFA with statistically significant results. The GTZAN in both experiments showed a smaller increase in performance using BCS and BDFA but results were found to be statistically insignificant. Overall, the results fail to prove this hypothesis.

Chapter 8
Conclusions and Further Work

The rise of large music databases is making automatically detecting the genre of music ever more important. The aim of this paper was to investigate the use of metaphor-based metaheuristic feature selection algorithms in music genre classification. Feature selection algorithms are used for a system to take large amounts of features which can then be filtered to provide the most relevant features. Metaphor-based metaheuristics offers a new set of feature selection algorithms. Metaphor-based metaheuristics and traditional feature selection algorithms were incorporated in two differing music genre classification systems to test the performance of the feature subsets they chose.

In Chapter 2, an overview of how machine learning is used in music genre classification was explored. Classification systems involve extracting features from a labelled dataset of features in order to build a statistical model of how these features correlate to genre. Predictions of unknown samples can be made by extracting the same features and comparing them to the model. Support vector machines were highlighted as ML algorithms that have seen successful results in music genre classification. Regardless of the algorithm used, the use of relevant features is required for these systems to be effective.

Following this, Chapter 3 discussed a range of features that can be employed for music genre classification. Features describing timbre, tonality and rhythm were presented. MFCC's and SSD were explored as features describing timbre. Tonal features extracted from Chromagrams were also explored. Rhythmic features using Rhythm Patterns were also explored. The number of potential features that could be extracted is limitless, but those presented are an example that have seen successful use in music genre classification.

In Chapter 4, an overview of feature selection algorithms and metaphor-based metaheuristics was provided. PCA and ReliefF-SFS were explored as current feature selection algorithms that have seen use in music genre classification. Self-Adaptive Harmony Search was identified as a recent feature selection algorithm based on the metaphor-based metaheuristic Harmony Search. This algorithm combined with selecting individual feature sets for binary classifiers was found to increase classification performance over a single feature set. Cuckoo Search and the Dragonfly Algorithm were identified as metaphor-based metaheuristics that have seen successful use in feature selection in classification problems but not music genre classification.

It was found that there is potential for metaphor-based metaheuristics to improve upon traditional feature selection algorithms in classification problems. Furthermore, using individual feature sets for binary classifiers was suggested to further improve the classification performance. It was hypothesised that these metaphor-based algorithms would outperform traditional feature selection algorithms, individual feature sets for binary classifiers would provide the strongest performance and BCS and BDFA would outperform SAHS. An experiment measuring the performance of traditional algorithms PCA and ReliefF-SFS against metaphor-based metaheuristics SAHS, BCA and BDFA was conducted. Results found BCA and BDFA to provide a slight increase in performance on the balanced GTZAN dataset. However, the ISMIR dataset produced opposing results which showed reduced performance when using the metaphor-based metaheuristics and a further drop when using them to create individual feature sets for binary classifiers. Overall, these results failed to prove these metaheuristics could outperform traditional feature selection algorithms or systems using no feature selection. There were some statistically significant results to suggest an improvement to results using BCS over SAHS and BDFA, but not enough to prove the hypothesis of BCS and BDFA outperforming SAHS.

This work was limited in the number of datasets tested. The results showed a clear difference in the performance of the algorithms using the two different datasets. Improvements with results in both experiments were found using the GTZAN dataset, but a decline in performance was found in the ISMIR dataset. Further research could explore how the dataset influences the feature selection algorithms. In this experiment it was suggested the difference in the datasets being balanced and unbalanced caused the difference in results. Further work could compare this or differing noisy datasets or datasets with differing number of classes.

In the case of the GTZAN dataset, the algorithms improved the performance of the results but not as much as hypothesised. This could potentially be due to the reduced number of iterations. As found from Huang et al., [2014], the iterations used was significantly lower to that of the literature. In this experiment it is suggested that the number of limitations limited the amount of available feature combinations that could be explored. Further research could reproduce this experiment with a greater number of iterations to identify the effect it has on classification performance.

The results also showed a significant lack of relevancy of rhythmic features in identifying genre. This may be due to genre being defined less by rhythm and more by timbre and tonality, or that the rhythmic features provided are ineffective descriptors of the rhythm of the sample. Further work can look at improving the relevancy of these rhythm features or identifying how much of musical genre is defined by rhythmic features.

Appendix

Tonal Features

Feature Description	Dimensionality	Overall Statistics	Total Number
CENS Chromagram	12	4**	48
Tonal Centroid	6	4*	24
Inharmonicity rate	1	1	1
HCDF	1	4*	4

Rhythmic Features

Feature Description	Dimensionality	Overall Statistics	Total Number
Rhythm Patterns	24	4*	96
Beat Spectrum	1	4*	4
Beat Strength	1	4*	4
Attack Time	1	4*	4
Pulse Clarity	1	4*	4
Tempo (AC)	1	1	1
Tempo (Spectrum)	1	1	1

Timbral Features

Feature Description	Dimensionality	Overall Statistics	Total Number
Spectral Centroid	1	4*	4
Spectral Bandwidth	1	4*	4
Spectral Contrast	1	4*	4
Spectral Flatness	1	4*	4
Spectrall Rolloff	1	4*	4
Zero Crossing Rate	1	4*	4
MFCC	13	4**	52
SSD	24	7	168

* mean, standard deviation, differential mean and differential standard deviation

** mean, standard deviation, skewness and kurtosis

References

Alpaydin, E. and Bach, F. (2014) *Introduction to Machine Learning*. Cambridge, United States: MIT Press.

Baniya, B.K. and Lee, J. (2016) 'Importance of audio feature reduction in automatic music genre classification', *Multimedia Tools and Applications*, 75(6), pp.3013–3026. doi: 10.1007/s11042-014-2418-z.

Barreira, L., Cavaco, S. and da Silva, J.F. (2011) 'Unsupervised Music Genre Classification with a Model-Based Approach', *Portuguese Conference on Artificial Intelligence*.

Bergstra, J. and Bengio, Y. (2012) 'Random Search for Hyper-parameter Optimization', *Journal of Machine Learning Research*, 13(1), pp.281–305.

Bishop, C.M. (2006) *Pattern recognition and machine learning*. New York: Springer.

Bowles, M. (2015) *Machine Learning in Python®: Essential Techniques for Predictive Analysis*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Chathuranga, D. and Jayaratne, D.L. (2013) 'Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches', *GSTF Journal on Computing (JoC)*, 3(2), pp.1–12. doi: 10.7603/s40601-013-0014-0.

D'Agostino, R. and Pearson, E.S. (1973) 'Tests for Departure from Normality. Empirical Results for the Distributions of b2 and √ b1', *Biometrika*, 60(3), pp.613–622. doi: 10.2307/2335012.

Fuhui, L., Ding, C. and Hanchuan, P. (2005) 'Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), pp.1226–1238. doi: 10.1109/TPAMI.2005.159.

Fulzele, P., Singh, R., Kaushik, N. and Pandey, K. (2018) 'A Hybrid Model for Music Genre Classification Using LSTM and SVM', *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Noida, India, 2-4 August. doi: 10.1109/IC3.2018.8530557.

Geem, Z.W., Kim, J.H. and Loganathan, G.V. (2001) 'A New Heuristic Optimization Algorithm: Harmony Search', *SIMULATION*, 76(2), pp.60–68. doi: 10.1177/003754970107600201.

Harte, C., Sandler, M. and Gasser, M. (2006) 'Detecting Harmonic Change in Musical Audio', *AMCMM '06*, Santa Barbara, California, USA, 27 October. doi: 10.1145/1178723.1178727.

Holzapfel, A. and Stylianou, Y. (2007) 'A Statistical Approach to Musical Genre Classification using Non-Negative Matrix Factorization', *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Honolulu, HI, USA, 15-20 April. doi: 10.1109/ICASSP.2007.366330.

Huang, Y.-F., Lin, S.-M., Wu, H.-Y. and Li, Y.-S. (2014) 'Music genre classification based on local feature selection using a self-adaptive harmony search algorithm', *Data & Knowledge Engineering*, 92(1), pp.60–76. doi: 10.1016/j.datak.2014.07.005.

Hunter, J.D. (2007) 'Matplotlib: A 2D Graphics Environment', *Computing in Science Engineering*, 9(3), pp.90–95. doi: 10.1109/MCSE.2007.55.

James, G. (2013) *An introduction to statistical learning: with applications in R*. New York: Springer.

Jones, E., Oliphant, T., Peterson, P. and others (2001) *SciPy: Open source scientific tools for Python*.

Kruspe, A., Lukashevich, H., Abeßer, J., Großmann, H. and Dittmar, C. (2011) 'Automatic Classification of Musical Pieces into Global Cultural Areas', *Audio Engineering Society Conference: 42nd International Conference: Semantic Audio*, Ilmenau, Germany, 22-24 July.

Lartillot, O., Toiviainen, P. and Eerola, T. (2008) 'A Matlab Toolbox for Music Information Retrieval'.

Li, T. and Tzanetakis, G. (2003) 'Factors in automatic musical genre classification of audio signals', *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, New Paltz, NY, USA, 19-22 October. doi: 10.1109/ASPAA.2003.1285840.

Lidy, T. and Rauber, A. (2005) 'Evaluation of Feature Extractors and Psycho-Acoustic Transformations for Music Genre Classification', *ISMIR 2005*, London, UK, 11-15 September.

Mafarja, M.M., Eleyan, D., Jaber, I., Hammouri, A. and Mirjalili, S. (2017) 'Binary Dragonfly Algorithm for Feature Selection', *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 11-13 October. doi: 10.1109/ICTCS.2017.43.

McFee, B., Raffel, C.A., Liang, D., Ellis, D.P.W., McVicar, M., Battenberg, E. and Nieto, O. (2015) 'librosa: Audio and Music Signal Analysis in Python'. doi: 10.25080/majora-7b98e3ed-003.

Medjahed, S.A., Ait Saadi, T., Benyettou, A. and Ouali, M. (2015) 'Binary Cuckoo Search Algorithm for Band Selection in Hyperspectral Image Classification', *IAENG International Journal of Computer Science*, 42(3), pp.1–9.

Mirjalili, S. (2016) 'Dragonfly Algorithm: A New Meta-heuristic Optimization Technique for Solving Single-objective, Discrete, and Multi-objective Problems', *Neural Comput. Appl.*, 27(4), pp.1053–1073. doi: 10.1007/s00521-015-1920-1.

Müller, M. and Ewert, S. (2012) 'Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features', *ISMIR 2011*, Miami, FL, USA, 24-28 October.

Müller, M., Kurth, F. and Clausen, M. (2005) 'Audio Matching via Chroma-Based Statistical Features', *ISMIR 2005*, London, UK, 11-15 September.

Murphy, K.P. (2012) *Machine Learning: A Probabilistic Perspective*. Cambridge, United States: MIT Press.

Ney, H. (1984) 'The use of a one-stage dynamic programming algorithm for connected word recognition', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2), pp.263–271. doi: 10.1109/TASSP.1984.1164320.

Oudre, L., Grenier, Y. and Fevotte, C. (2011) 'Chord Recognition by Fitting Rescaled Chroma Vectors to Chord Templates', *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7), pp.2222–2233. doi: 10.1109/TASL.2011.2139205.

Pang, B., Song, Y., Zhang, C., Wang, H. and Yang, R. (2018) 'A Modified Artificial Bee Colony Algorithm Based on the Self-Learning Mechanism', *Algorithms*, 11(6), pp.78. doi: 10.3390/a11060078.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2011) 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, 12(1), pp.2825−2830.

Rajanna, A.R., Aryafar, K., Shokoufandeh, A. and Ptucha, R. (2015) 'Deep Neural Networks: A Case Study for Music Genre Classification', *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, USA, 9-11 December. doi: 10.1109/ICMLA.2015.160.

Rodrigues, D., Pereira, L.A.M., Almeida, T.N.S., Papa, J.P., Souza, A.N., Ramos, C.C.O. and Yang, X. (2013) 'BCS: A Binary Cuckoo Search algorithm for feature selection', *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, Beijing, China, 19 May. doi: 10.1109/ISCAS.2013.6571881.

Rosner, A. and Kostek, B. (2018) 'Automatic music genre classification based on musical instrument track separation', *Journal of Intelligent Information Systems*, 50(2), pp.363–384. doi: 10.1007/s10844-017-0464-5.

Schubert, E., Wolfe, J. and Tarnopolsky, A. (2004) 'Spectral centroid and timbre in complex, multiple instrumental textures', *8th International Conference on Music Perception & Cognition (ICMPC)*, Chicago, USA, 3-7 August.

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

Shalev-Shwartz, S. and Ben-David, S. (2014) *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press.

Shao, X., Xu, C. and Kankanhalli, M.S. (2004) 'Unsupervised classification of music genre using hidden markov model', *2004 IEEE International Conference on Multimedia and Expo (ICME)*, Taipei, Taiwan, 27-30 June.

Sharma, S., Fulzele, P. and Sreedevi, I. (2018) 'Novel hybrid model for music genre classification based on support vector machine', *2018 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, Penang, Malaysia, 28-29 April. doi: 10.1109/ISCAIE.2018.8405505.

Stein, M., Schubert, B.M., Gruhne, M., Gatzsche, G. and Mehnert, M. (2009) 'Evaluation and Comparison of Audio Chroma Feature Extraction Methods', *126th Audio Engineering Society Convention*, Munich, Germany, 7-10 May.

Stevens, S.S., Volkmann, J. and Newman, E.B. (1937) 'A Scale for the Measurement of the Psychological Magnitude Pitch', *The Journal of the Acoustical Society of America*, 8(3), pp.185–190. doi: 10.1121/1.1915893.

Tsunoo, E., Tzanetakis, G., Ono, N. and Sagayama, S. (2011) 'Beyond Timbral Statistics: Improving Music Classification Using Percussive Patterns and Bass Lines', *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), pp.1003–1014. doi: 10.1109/TASL.2010.2073706.

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

Webb, A.R. (2003) *Statistical Pattern Recognition*: John Wiley & Sons.

Wu, J., Gupta, S. and Bajaj, C. (2016) 'Higher Order Mutual Information Approximation for Feature Selection', *Computing Research Repository*, abs/1612.00554.

Wu, M. and Wang, Y. (2015) 'A feature selection algorithm of music genre classification based on ReliefF and SFS', *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, Las Vegas, NV, USA, 28-1 June/July. doi: 10.1109/ICIS.2015.7166651.

Yang, X.-S. and Suash Deb (2009) 'Cuckoo Search via Levy flights', *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, 9 December. doi: 10.1109/NABIC.2009.5393690.

Yaslan, Y. and Cataltepe, Z. (2006) 'Audio Music Genre Classification Using Different Classifiers and Feature Selection Methods', *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, 20-24 August. doi: 10.1109/ICPR.2006.282.

Zwicker, E. (1961) 'Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen)', *The Journal of the Acoustical Society of America*, 33(2), pp.248–248. doi: 10.1121/1.1908630.